НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

«На правах рукопису» «До захисту допущено»

УДК _____ В.о. завідувача кафедри

_____ М.В.Грайворонський

“___”_____2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності: 113 Прикладна математика

на тему: Машинне навчання для аналізу тональності тексту на прикладі передвиборчих президентських перегонів

Виконав : студент 2 курсу, групи ФІ-72мп
(шифр групи)

Павленко Павло Ігорович
(прізвище, ім’я, по батькові) (підпис)

Науковий керівник д.т.н. Качинський А.Б.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою
Спеціальність (спеціалізація) – 113 Прикладна математика («Аналітичні методи безпеки інформації»)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ М.В.Грайворонський
(підпис)

«____»_____2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Павленко Павлу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема дисертації :Машинне навчання для аналізу тональності тексту на прикладі передвиборчих президентських перегонів

науковий керівник дисертації: Качинський Анатолій Броніславович, доктор технічних наук
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «____ » _____ 2018 р. № _____

2. Термін подання студентом дисертації 10 грудня 2018 р.

3. Об'єкт дослідження : думки в ЗМІ та думки користувачів щодо перемоги одного із кандидатів у президентських виборах

4. Вихідні дані : тональність думок в ЗМІ та тональність думок користувачів щодо одного із кандидатів у президентських виборах

5. Перелік завдань, які потрібно розробити : a. Провести огляд існуючих літературних джерел щодо методів класифікації у машинному навчанні;

b. Ознайомлення з пакетом програм scikit-learn;

c. Розробка програмної реалізації методів машинного навчання (логістична регресія, метод k-найближчих сусідів, метод опорних векторів, наївний байесів класифікатор, дерево рішень, алгоритм Random Forest);

.

6. Орієнтовний перелік ілюстративного матеріалу : 12 -15 ілюстрацій

7. Орієнтовний перелік публікацій

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 03.09.2018

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Опрацювання літературних джерел	30.09.2018	Виконано
2	Отримання технічного завдання	01.10.2018	Виконано
3	Ознайомлення з принципами програмування Python	10.10.2018	Виконано
4	Розробка класифікаторів машинного навчання	15.11.2018	Виконано
5	Написання магістерської дисертації	03.12.2018	Виконано
6	Підготовка до захисту	18.12.2018	Виконано

Студент _____ _____
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ _____
(підпис) (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника магістерської дисертації.

РЕФЕРАТ

Робота обсягом 76 сторінок включає 23 ілюстрацій, 12 таблиць і 19 джерел літератури.

Метою даної роботи є розробка і використання методики, яка дозволить здійснювати аналіз тональності тексту.

Об’єктом дослідження є думки у засобах масової інформації та думки користувачів, щодо перемоги одного із кандидатів у президентських виборах..

Предметом дослідження є тональність думок в засобах масової інформації та тональність думок користувачів, щодо одного із кандидатів у президентських виборах.

В процесі виконання даної дипломної роботи детально вивчені існуючі методи машинного навчання, алгоритми класифікації, методи аналізу тональності тексту. Запропонована методика аналізу тональності тексту для ситуації, коли потрібно спрогнозувати переможця президентських виборів. Розглянута ситуація другого туру президентських виборів, у якості прикладу застосування такої методики.

Результати даного дослідження можуть бути використані для прогнозування переможця у президентських виборах, аналізу думок виборців щодо кандидатів у президентських перегонах.

.

ABSTRACT

Work consists of 76 pages that includes 23 illustrations, 12 tables and 19 sources of literature.

The purpose of this work is to develop and use a technique that will allow you to analyze the tonality (sentiment analysis) of the text.

The subject of the research is the opinion of the media and the views of users about the victory of one of the candidates in the presidential election.

The subject of the study is the tone of thought in the media and the tone of user opinions about one of the candidates in the presidential election.

In the course of this thesis, the existing methods of machine learning, classification algorithms, methods for analyzing the tonality of the text (sentiment analysis) are studied in detail. The method of analysis of the tone of the text for the situation when it is necessary to predict the winner of the presidential election is proposed. The situation of the second round of presidential elections are considered, as an example of the application of such a method.

The results of this study can be used to predict the winner in the presidential election, to analyze the views of voters about candidates in the presidential race.

.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 8
Вступ 10
1 Теоретичні відомості про методи машинного навчання, їх обгрунтування 12
1.1 Огляд джерел за тематикою, що задана 12
1.2 Логістична регресія 14
1.3 Метод k-найближчих сусідів 16
1.4 Метод опорних векторів (SVM) 18
1.5 Наївний баєсів класифікатор 24
1.6 Дерево ухвалення рішень 26
1.7 Алгоритм Random Forest 29
1.8 Перевірка результатів роботи алгоритмів 32
Висновки до розділу 1 33
2 Математична модель “мішок слів” (“Bag of words”) 34
2.1 Модель “мішок слів” 34
2.2 Модель “мішок термів” 35
2.3 Частотна модель тексту 35
2.4 Векторна модель тексту 36
2.5 Частотний словник як векторна модель тексту 37
2.6 Латентно-семантичний аналіз 38
2.7 Нормалізація 39
2.8 Стемінг 40
Висновки до розділу 2 41
3 Аналіз результатів досліджень та проведених соціологічних опитувань 42
3.1 Моніторинг електоральних настроїв українців (Київський міжнародний інститут соціології) 42
3.2 Соцопитування MPI (Міжнародний республіканський інститут): передвиборчі настрої в Україні 53
3.3 Результати соцопитування проведеного КМІС, Центром Разумкова та СОЦИС. 55
3.4 Результати алгоритмів машинного навчання (класифікації) на власній вибірці даних. Власний прогноз 62
Висновки до розділу 3 64
4 Розроблення стартап-проекту 65

.

4.1 Опис ідеї проекту 65
4.2 Технологічний аудит ідеї проекту 66
4.3 Аналіз ринкових можливостей для запуску стартап-проекту 66
4.4 Розроблення ринкової стратегії проекту 68
4.5 Розроблення маркетингової програми стартап-проекту 69
Висновки до розділу 4 70
Висновки 71
Перелік джерел 72
Додаток А 74

.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Аналіз тональності тексту (сентимент-аналіз, англ. Sentiment analysis, англ. Opinion mining) -в комп'ютерній лінгвістиці це підмножина методів аналізу змісту текстів, головна задача яких це оцінка емоцій авторів думок відносно об’єктів чи подій, автоматичне виявлення типової лексики, яка надає тексту відтінок певної емоції.

Тональність - це висловлення автором тексту емоції по відношенню до деякого об'єкту, суб’єкта, реальної події, що виражене в тексті. Тональність всього тексту виражається як функція (в спрощеному випадку відбувається сумування) лексичних тональностей складових його одиниць (пропозицій) іззастосуванням правил, що поєднують ці лексичні одиниці.

Машинне навчання (англ. machine learning) — це галузь інформатики в області штучного інтелекту. Основна задача – застосувати статистичні та математичні методи та прийоми для надання комп'ютерам здатності «навчатися» (тобто вирішувати задачу більш ефективно по відношенню до певного критерію) з даних. Навчання має відбуватися самою програмою, а не бути заданим у коді програмістом.

Наука про дані — це міждисциплінарна галузь про наукові методи, процеси і системи, які стосуються добування знань із даних у різних формах, як структурованих так і неструктурованих. Наука про дані є продовженням деяких галузей аналізу даних, таких як статистика, класифікація, кластеризація, машинне навчання, добування даних і передбачувальна аналітика.

Обробка природної мови (англ. Natural-language processing, NLP) — це міждисциплінарна наука в галузі інформатики, штучного інтелекту та математики. Вона вивчає проблеми комп'ютерного аналізу та синтезу природної мови. Стосовно штучного інтелекту аналіз означає розуміння мови, а синтез — генерацію розумного тексту. Вирішення цих задач означатиме більш результативну і простішу взаємодію комп’ютерів та людей.

.

Кортеж - упорядкований кінцевий набір довжини n (де $n \in N \cup \{0\}$), кожен з елементів якого fi належить деякій множині F. Елементи кортежу називаються його компонентами, або координатами. Елементи кортежу можуть повторюватися в ньому будь-яке число раз. Багато математичних об'єктів формально визначаються як кортежі. Точка в n-вимірному просторі дійсних чисел визначається як кортеж довжини n, складений з елементів безлічі дійсних чисел.

Вектор - кортеж однорідних елементів (скалярів). Сукупність векторів утворює векторний простір V над лінійним простором (полем) F. Елементи V називаються векторами, елементи F - скалярами.

Стоп-символи (стоп-слова, шумові слова) - це слова, що зустрічаються практично у всіх текстах і не несуть спеціальної смислового навантаження. У більшості алгоритмів обробки текстів, ці слова ігноруються (зокрема, пошуковими машинами при індексуванні).

.

ВСТУП

Аби вивчити основні напрями та зміни виборчих установок, поведінки виборців у електоральній науковій соціології та реалізації функцій соціології використовуються різні види та методи дослідження, соціологічного прогнозування. Наприклад методи масового та експертного оцінювання, передвиборчі оцінки, екзит-поли в день голосування (exit poll) та післявиборчі, базові, рейтингові, іміджево-рекламні тощо.

Головними серед них є наступні:

I. Найбільш розповсюджені серед інших видів електоральних досліджень набули рейтингові електоральні дослідження, метою яких є виявлення шансів партій та кандидатів на певних типах виборів. Для цього проводять передвиборчі та післявиборчі електоральні опитування. Такі опитування проводять аби дізнатися думку простих виборців та зробити прогноз.

II. Базові соціологічні дослідження передвиборчої ситуації. Для цих досліджень головною метою і задачею яких є, як зазначають О. Петров та В. Полторак, збір та обробка інформації для виборчої кампанії, аби можна було провести стратегічне планування.

III. Опитування в день голосування (Exit poll) – це короткі опитування, що проводяться в день голосування на місці волевиявлення, одразу після акту голосування. Основним завданням екзит-полів є вивчення факторів реального вибору людей, що прийшли на виборчі дільниці, та контроль за результатами волевиявлення.

IV. Передвиборчі та післявиборчі рекламні та іміджеві опитування. Головною метою таких досліджень є виявлення іміджу кандидатів та партій у масовій свідомості виборців, того на скільки ефективно працюють різних види передвиборчих рекламних кампаній партій та кандидатів у ЗМІ, масових акцій на підтримку одного із кандидатів, соціальних та адміністративних технологій.

.

За останнє десятиліття політичні команди кандидатів у президенти почали використовувати технології аналізу великих даних не тільки для прогнозування результатів, але і для отримання додаткової інформації про електорат. Згодом цю інформацію можна вміло конвертувати у боротьбу за ту частину електорату, яка ще не визначилася із вибором одного з учасників президентських перегонів. Технології науки про дані, аналізу даних та обробки великих даних вдало були застосовані при президентських перегонах в США у 2008, 2012 та 2016 роках і допомогли перемогти Дональду Трампу та двічі Бараку Обамі. Аналогічні тенденції слід очікувати і для президентських передвиборчих перегонів в Україні у 2019 році.

Актуальність роботи – зумовлена тим, що в умовах російської агресії по відношенню до України будь-який виборчий процес може використовуватись для дестабілізації політичної та економічної ситуації в Україні.

Завдання дослідження – здійснити порівняльний аналіз алгоритмів машинного навчання для вирішення задачі класифікації на прикладі ситуації другого туру передвиборчих президентських перегонів.

Мета дослідження – розробити і використати методику, яка дозволить здійснювати аналіз тональності тексту

Об’єкт дослідження – думки у засобах масової інформації та думки користувачів, щодо перемоги одного із кандидатів у президентських виборах.

Предмет дослідження – тональність думок в засобах масової інформації та тональність думок користувачів, щодо одного із кандидатів у президентських виборах.

Методи дослідження – алгоритми класифікації машинного навчання, а саме логістична регресія, метод к-найближчих сусідів, метод опорних векторів, наївний Байес, дерево рішень, алгоритм “Random Forest”.

Практичне значення одержаних результатів – за допомогою отриманих у даній дипломній роботі результатів можна спрогнозувати переможця другого туру президентських виборів 2019 року в Україні.

.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО МЕТОДИ МАШИННОГО НАВЧАННЯ, ЇХ ОБГРУНТУВАННЯ

Для розв’язання задачі класифікації емоцій в одну з категорій (позитивна або негативна), тобто аналізу тональності тексту, були використані нижченаведені алгоритми машинного навчання.

1.1 Огляд джерел за тематикою, що задана

1) Єндрю Брюс, Пітер Брюс “Practical Statistics for Data Scientists”

Книга розрахована на фахівців в області Data Science, що володіють деяким досвідом роботи з мовою програмування R і мають попереднє поняття про математичну статистику. У книзі в зручній і доступній формі представлені ключові поняття з статистики, які відносяться до науки про дані, а також пояснено, які поняття важливі і корисні з точки зору науки про дані, які менш важливі і чому. Докладно розкрито теми: розвідувальний аналіз даних, розподілу даних і вибірок, статистичні експерименти і перевірка значущості, регресія і передбачення, класифікація, статистичне машинне навчання і навчання без учителя.

2) Джузеппе Бонаккорсо “Machine Learning Algorithms”

Оскільки обсяг даних продовжує зростати з майже нестримною швидкістю, здатність розуміти та обробляти дані стає ключовою відмінністю для конкуруючих організацій. Програми для машинного навчання знаходяться скрізь, від виявлення спаму, пошуку документів та стратегій торгівлі, до розпізнавання мови. Це робить машинне навчання добре пристосованим до сучасної епохи "Великих даних і наукових даних". Головним завданням є те, як перетворити дані в знання.

У цій книзі викладені всі важливі алгоритми машинного навчання, які широко використовуються в галузі науки про дані. Ці алгоритми можуть бути використані для навчання з учителем та без учителем, навчання у середовищі. Кілька знаменитих алгоритмів, які розглядаються в цій книзі, є лінійна регресія, логістична регресія, SVM, Наївний Байєс, K-Means, Random Forest, TensorFlow та

Feature Engineering. У цій книзі також викладено, як працюють ці алгоритми та їх практичне застосування для вирішення задач. Ця книга також знайомить з системами природної обробки мов та рекомендацій, які допоможуть одночасно запускати декілька алгоритмів, описано вибір алгоритмів машинного навчання для кластеризації, класифікації або регресії на основі конкретної проблеми.

3) А.Б. Качинський, С.П. Герасимчук, Д.І. Остапчук, Л.М. Шипілова “Визначення місця і ролі національної єдності у Стратегії національної безпеки України, програмах політичних партій і передвиборчому процесі 2007 року”

У книзі застосовані та описані новітні методи та враховані традиційні показники ваги цілі “забезпечення національної єдності” для акторів політичної системи України, за допомогою яких досліджено стан національної єдності як цілі Стратегії національної безпеки для політичних акторів. Для отримання необхідних даних була задіяна інформаційно-аналітична система “АРКС”, завдяки якій були отримані оцінки актуальності цілі, її відносної важливості, рівня політизації, розбіжності думок політичних акторів. Книга розрахована на широке коло науковців, політиків, державних службовців і всіх, хто цікавиться проблемами національної безпеки України.

4) Хобсон Лейн, Коул Ховард, Ханс Хапке “Natural Language Processing in Action”

У книзі описані методи та підходи обробки природних мов (NLP). NLP - це дисципліна для навчання комп’ютерів розуміння та обробки людських мов. Приклади застосування NLP ви можете побачити всюди, починаючи від чатботів і до програмного забезпечення розпізнавання мовлення на вашому телефоні. Сучасні методи NLP засновані на машинному навчанні, радикально покращують здатність програмного забезпечення розпізнавати шаблони, використовують контекст для виведення сенсу та точного розпізнавання намірів з погано структурованого тексту. NLP обіцяє допомогти покращити взаємодію з клієнтами, заощадити витрати та покращити програми такі як пошук або підтримка продуктів.

.

У книзі викладені наступні практичні поради: Робота з Keras, TensorFlow, Gensim, scikit-learn та багато іншого; Розбір та нормалізація тексту; Нормативні (граматичні) NLP; Дані (машинного навчання) NLP; Глибоке навчання NLP; Алгоритми оптимізації гіперпараметрів.

1.2 Логістична регресія

Логістична регресія (англ. logistic regression) - статистичний регресійний метод. Логістичну регресію дослідники застосовують у задачах класифікації, у випадку, коли залежна змінна представляється різними категоріями, тобто може набувати тільки скінченної множини значень (наприклад 0 та 1).

Якщо порівнювати із звичайною регресійною моделлю, то метод логістичної регресії не виконує прогноз для значень числової змінної виходячи з вибірки вихідних значень незалежної змінної. Замість цього, значеннями для функції є ймовірність того, що дане вихідне значення належить до одного із класів. Аби спростити задачу припустимо, що у нас є тільки два класу і ймовірність, яку ми будемо визначати, вірогідності того, що певне значення належить класу "+". Таким чином, результат логістичної регресії завжди буде знаходитись в інтервалі [0, 1], як і ймовірність.

Головна суть логістичної регресії полягає в тому, що простір вихідних значень незалежної змінної може бути розділений лінійною площиною (тобто, прямою) на два класи. Якщо точніше, то під лінійною площиною при двох параметрах (два виміри) мають на увазі пряму лінію, що розділяє класи (без вигинів). У випадку трьох вимірів — це площини, і так далі. Ця межа визначається в залежності від наявних вихідних даних та навчального алгоритму. Щоб все працювало, точки вихідних даних повинні розділятися лінійною площиною на дві вищезазначених області. Якщо точки вихідних даних задовольняють цій вимозі, то їх можна назвати лінійно роздільними. Візуально це зображено на на Рисунку 1.1.

.

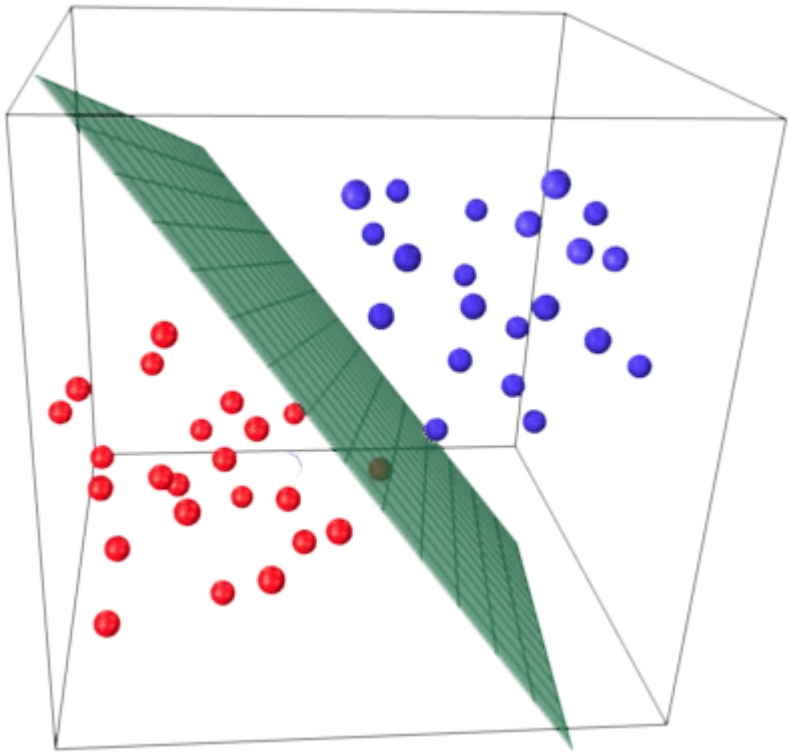

Рисунок 1.1- Візуалізація класифікатора

Зазначена на рисунку площина називається лінійним дискримінантом. Вона є лінійною з точки зору своєї функції і моделі, а також проводить поділ, якщо точніше то дискримінацію точок на різні класи. Якщо неможливо провести лінійний розподіл точок у вихідному просторі або вони взагалі лінійно нероздільні, то варто спробувати перетворити вектори ознак в простір більшої розмірності, додавши додаткові ефекти взаємодії, члени більш високого ступеня та інше. Такий метод називають ”трюк з ядром” (Kernel trick). Використання лінійного алгоритму в такому випадку дає певні переваги для навчання нелінійної функції, оскільки межа стає нелінійною при поверненні у вихідний простір.

Визначення логістичної моделі

Нехай є деяка випадкова величина Y, що може набувати лише двох значень. Ці значення позначаються цифрами 0 і 1 (2 класи). Припустимо, що ця величина залежить від деякої множини незалежних змінних $x = (1, x_1, \dots, x_n)^T$. Залежність Y від $x_1, \dots, x_n$ можна визначити ввівши додаткову змінну y*, де y*= $\theta^T = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n + \varepsilon$ Тоді:

$$Y = \begin{cases} 0, & y^* \leqslant 0 \\ 1, & y^* > 0 \end{cases}$$

При визначенні логістичної моделі стохастичний доданок ε вважається випадковою величиною з логістичним розподілом ймовірностей. Тому можна зробити висновок, що для певних конкретних значень змінних $x^* = x_1^*, \ldots, x_n^*$ одержується відповідне значення y* і ймовірність того, що Y=1 є наступною:

$$p(Y = 1) = p(y^* > 0) = p(\theta^T x^* + \varepsilon > 0) = p(\varepsilon > -\theta^T x^*) = p(\varepsilon \leqslant \theta^T x^*) = \Lambda(\theta^T x^*).$$

Передостання рівність випливає з симетричності логістичного розподілу,
Λ позначає логістичну функцію — функцію розподілу логістичного розподілу:

$$\Lambda(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}$$

Тому можна сказати, що для конкретного значення x^i випадкова величина Y^i має розподіл Бернуллі: $Y^i \sim B(1, \Lambda(\theta^T x^i))$

Логістична модель задовольняє наступній умові [6]:

$$\ln \frac{p(1|X)}{1 - p(1|X)} = \ln \frac{p(1|X)}{p(0|X)} = b_0 + b_1 x_1 + \ldots + b_J x_J$$

1.3 Метод k-найближчих сусідів

Метод k-найближчих сусідів — це алгоритм машинного навчання для автоматичної класифікації об'єктів, заснований на метричних показниках. Основна суть методу найближчих сусідів полягає в тому, що об’єкту присвоюється клас, який є найбільш поширеним серед сусідніх елементів даного елемента. Сусіди беруться, виходячи з множини об'єктів, класи яких уже відомі, і за допомогою гіперпараметру k. Цей гіперпараметр вираховує, який клас є найчисленнішим серед сусідів. Кожен об'єкт має кінцеву кількість атрибутів. Це алгоритм навчання з учителем, тому для роботи алгоритму потрібна розмічена вибірка даних.

Метод k-найближчих сусідів (англ. k-nearest neighbor method) — це метод класифікації даних, що є непараметричним. Для вирішення задачі класифікації об'єктів у рамках простору властивостей використовуються різні відстані
.

(евклідові, відстань Мінковського), пораховані до усіх інших об'єктів. Вибираються об'єкти, до яких відстань найменша, і вони виділяються в окремий клас. Приклад роботи класифікатора наведено на Рисунку 1.2.

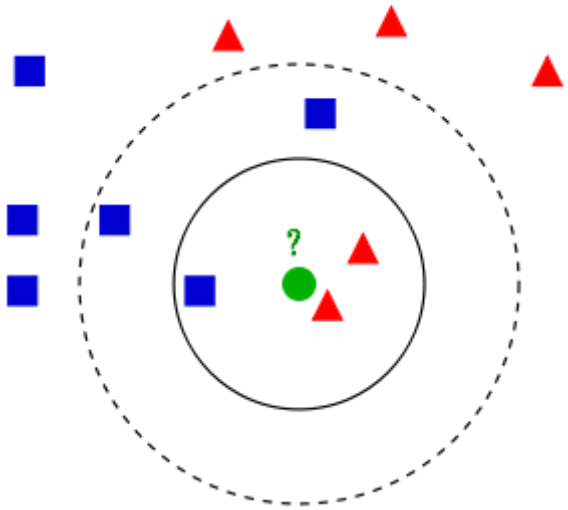

Рисунок 1.2- Візуалізація класифікатора

Вибірка складається з двох класів: синій квадрат (клас 1) та червоний трикутник (клас 2). Потрібно класифікувати зелене коло до одного з цих класів. Якщо гіперпараметр k = 3, то зелене коло буде віднесено до другого класу, тому що всередині меншого кола два трикутника і тільки один квадрат. Якщо гіперпараметр k = 5, то коло буде віднесено до другого класу, бо всередині більшого кола три квадрата проти двох трикутників[7].

Алгоритм може бути застосований до вибірок з великою кількістю атрибутів (багатовимірним). Для цього перед застосуванням потрібно визначити функцію дистанції. Класичний варіант визначення дистанції - дистанція в евклідовому просторі.

Нормалізація

Однак різні атрибути можуть мати різний діапазон представлених значень у вибірці (наприклад атрибут А представлений в діапазоні від 0.1 до 0.5, а атрибут Б представлений в діапазоні від 1000 до 5000), то значення дистанції можуть сильно залежати від атрибутів з великими діапазонами. Тому дані зазвичай підлягають нормалізації. При кластерному аналізі є два основних способи нормалізації даних:

Міні-макс нормалізація:

$$x' = (x - MIN[X])/(MAX[X] - MIN[X])$$

В цьому випадку всі значення будуть лежати в діапазоні від 0 до 1. Дискретні бінарні значення визначаються як 0 і 1.

Z-нормалізація:

$$x' = (x - M[X])/\sigma[X]$$

Виділення значущих атрибутів

Деякі значущі атрибути можуть бути важливіше інших, тому для кожного атрибута може бути заданий у відповідність певне політичне значення (наприклад обчислений за допомогою тестової вибірки і оптимізації помилки відхилення). Таким чином, кожному атрибуту k буде поставлено у відповідність вага z_k, так що значення атрибута буде потрапляти в діапазон $[0; z_k \max(k)]$ (для нормалізованих значень за методом міні-макс). Наприклад, якщо атрибуту присвоєно вагу 2.7, то його нормалізувати-зважене значення буде лежати в діапазоні [0; 2.7].

Визначення класу

При такому способі до уваги береться не тільки кількість потрапили в область певних класів, а й їх віддаленість від нового значення.

Для кожного класу j визначається оцінка близькості:

$$Q_j = \sum_{i=1}^{n} \frac{1}{d(x, a_i)^2},$$

Тут d(x, a) - дистанція від нового значення x до об'єкта a.

У якого класу вище значення близькості, той клас і присвоюється новому об'єкту [8].

1.4 Метод опорних векторів (SVM)

Метод опорних векторів (англ. SVM, support vector machine) - набір схожих алгоритмів навчання з учителем, що використовуються для задач класифікації та регресійного аналізу. Належить сімейству лінійних класифікаторів і може також

розглядатися як спеціальний випадок регуляризації по Тихонову. Особливою властивістю методу опорних векторів є невпинне зменшення емпіричної помилки класифікації і збільшення відстані (margin), тому метод також відомий як метод класифікатора з максимальною відстанню. Принцип роботи наступний: Враховуючи набір навчальних прикладів, кожен з яких позначається як такий, що належить до тієї чи іншої з двох категорій, алгоритм навчання SVM створює модель, яка призначає нові приклади однієї категорії або іншої, що робить його неімовірнисним бінарним лінійним класифікатором (хоча методи наприклад, масштабування Platt існує для використання SVM в імовірнісному класифікації). SVM-модель являє собою представлення прикладів як точок в просторі, закріплені таким чином, що приклади окремих категорій ділиться явним розділенням, яке є максимально можливим. Нові дані потім вкладаються в той самий простір і передбачають, що вони належать до категорії, на підставі яких була сформована тренувальна вибірка. Крім виконання лінійної класифікації, SVM може ефективно виконувати нелінійну класифікацію, використовуючи те, що називається ”трюком з ядром” (kernel trick), що неявно відображає їхні входи у високорозмірних просторах.

Коли дані не є розміченими, навчання з учителем неможливе, і виникає необхідність у застосування спонтанного навчання, яке намагається знайти природну кластеризацію даних груп, а потім класифікувати нові дані до цих сформованих груп. Тобто, фактично, вирішується спочатку задача кластеризації одним із методів (метод к-середніх, метод ієрархічного кластерування), а вже потім безпосередньо задача класифікації. Алгоритм векторної кластеризації, створений Хавою Сігельманом та Володимиром Вапніком, застосовує статистику векторів підтримки, розроблених в алгоритмі векторних машин підтримки, для класифікації нерозмічених даних і є одним з найбільш широко використовуваних алгоритмів кластеризації в промислових цілях. Приклад роботи класифікатора наведено на Рисунку 1.3.

.

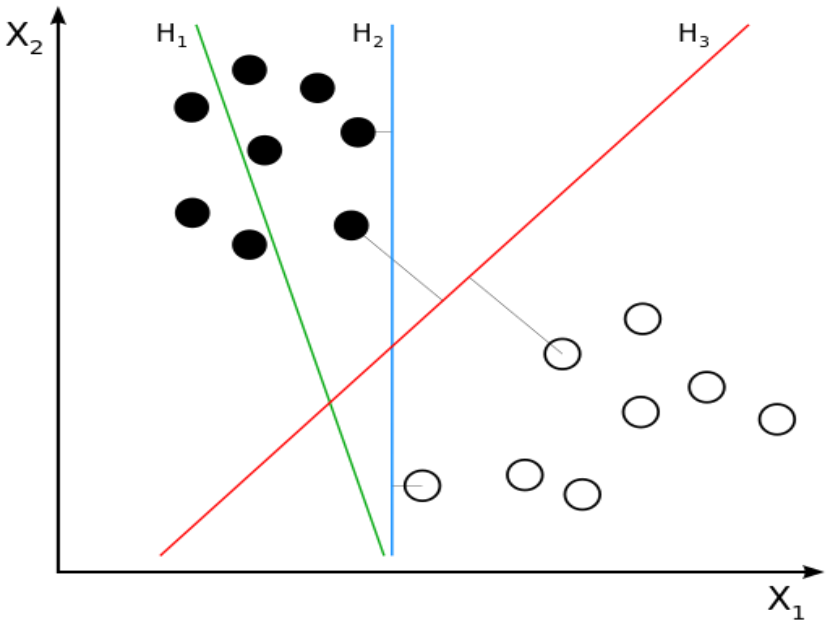

Рисунок 1.3- Візуалізація класифікатора

Гіперплощина H1 не є роздільною. Гіперплощина H2 є роздільною, але не з максимальним розділенням. H3 є роздільною гіперплощиною із максимальним розділенням.

Постановка задачі

Класифікація даних - це важливе завдання в галузі машинного навчання. Припустимо, що деякі задані точки в заданому просторі певної розмірності належать до одного з двох класів, а метою класифікації є визначення того, до якого класу потрапить нова точка з вибірки даних. У випадку алгоритму опорних векторів точка даних розглядається як p-мірний вектор (список з p чисел), і ми хочемо знати, чи можемо ми розділити ці точки (p-1) –мірною роздільною гіперплощиною. Цю площину називають лінійним класифікатором. Є багато гіперплощин, які можуть класифікувати дані. Можна стверджувати, що найкраща гіперплоща - це та площина, яка проводить собою найбільше розділення між двома класами. Таким чином, ми вибираємо гіперплощину так, щоб максимальна була відстань від неї до найближчої точки даних на кожній стороні. Якщо така гіперплощина існує, вона відома як гіперплощина з максимальним розділенням, а

.

лінійний класифікатор, який її визначає, відомий як максимальний роздільний класифікатор; або як, персептрон оптимальної стійкості.

Визначення

Більш формально SVM створює гіперплощину або набір гіперплощин у високому або нескінченновимірному просторі, які можуть бути використані для класифікації, регресії або інших завдань, таких як виявлення викидів аномалій. Інтуїтивно добрий розподіл досягається гіперплощиною, яка має найбільшу відстань до найближчої точки навчальних даних будь-якого класу (так звана функціональна маржа), так як загалом, чим більший роздільний запас, тим нижча похибка узагальнення класифікатора.

В той час, як первинну задачу може бути сформульовано у скінченно вимірному просторі, часто трапляється так, що множини, які треба розрізнювати, не є лінійно роздільними в ньому. З цієї причини було запропоновано відображувати первинний скінченно вимірний простір до простору набагато вищої вимірності, здогадно роблячи розділення простішим у тому просторі. Для збереження помірного обчислювального навантаження, відображення, які використовуються методом опорних векторів, розробляють такими, щоби забезпечувати можливість простого обчислення скалярних добутків у термінах змінних первинного простору, визначаючи їх у термінах ядрових функцій k(x,y) , що їх обирають відповідно до задачі. Гіперплощини в просторі вищої вимірності визначаються як геометричне місце точок, чиї скалярні добутки з вектором у цьому просторі є сталими. Вектори, які визначають гіперплощини, можуть обиратися як лінійні комбінації з параметрами α_i відображень векторів ознак x_i , які трапляються в базі даних. За такого вибору гіперплощини, точки x простору ознак, які відображаються на гіперплощину, визначаються відношенням $\sum_i \alpha_i k(x_i, x) = const$. Зауважте, що якщо k(x,y) стає малою з віддаленням у від x , то кожен член цієї суми вимірює ступінь близькості пробної точки x до відповідних основних точок даних x_i .

.

Таким чином, сума вищевказаних ядер може бути використана для вимірювання відносної близькості кожної тестової точки до точок даних, що виникають з одного чи іншого з наборів, які підлягають дискримінації. Зверніть увагу на те, що набір точок x, перерахованих до будь-якої гіперплощини, може бути досить заплутаним і складним у результаті. Тобто, SVM дозволяє виконати більш складну дискримінацію між множинами, які взагалі не опуклі в початковому просторі.

Застосування

SVM може бути використаний для вирішення різних реальних проблем:

- SVM корисний в текстовій та гіпертекстовій класифікації, оскільки його застосування може значно зменшити потребу в маркованих навчальних примірниках або у стандартних індуктивних, так і в трансдуктівних параметрах.
- Класифікація зображень також може бути виконана за допомогою SVM. Експериментальні результати показують, що SVM досягає значно вищої точності пошуку, ніж традиційні схеми уточнення запитів після трьох-чотирьох раундів зворотного зв'язку. Це також стосується систем сегментації зображень, включаючи ті, що використовують модифіковану версію SVM, яка використовує привілейований підхід, як це запропонував Вапнік.
- Рукописні символи можуть бути розпізнані за допомогою SVM
- Алгоритм SVM широко застосовується в біології та інших науках. Вони використовувались для класифікації білків, до 90% правильно класифікують сполуки. Перестановні тести на основі SVM були запропоновані як механізм інтерпретації моделей SVM. Підтримка векторних машинних ваг також використовувалася для інтерпретації моделей SVM в минулому. Пост-інтерпретація векторних машинних моделей підтримки з метою виявлення ознак, що використовуються моделлю для прогнозування, є відносно новою областю досліджень, що мають особливе значення в біологічних науках.

.

Лінійна SVM

В нас є тренувальний набір даних з n точок вигляду:

$(\overrightarrow{x_1}, y_1), \ldots, (\overrightarrow{x_n}, y_n)$

Тут y_i є або 1, або -1, і кожна з цих точок вказує нам клас, до якого належить точка $\overrightarrow{x_i}$. Кожна $\overrightarrow{x_i}$ є p-вимірним дійсним вектором. Задача описується як задача класифікації, де нам потрібно вірно відкласифікувати вхідні дані. Для цього потрібно знайти максимальну розділову гіперплощину, яка відділяє групу точок $\overrightarrow{x_i}$, де y_i =1 , від групи точок, для яких y_i = -1 , і визначається таким чином, що відстань між цією гіперплощиною та найближчою точкою $\overrightarrow{x_i}$ з кожної з груп є максимальною.

Ми знаємо, що будь-яку гіперплощину можна записати як множину точок $\vec{x}$, які задовольняють рівності $\vec{\omega} \cdot \vec{x} - b = 0$, де $\vec{\omega}$ є (не обов'язково нормалізованим) вектором нормалі до цієї гіперплощини. Параметр $\frac{b}{||\vec{\omega}||}$ визначає зсув гіперплощини від початку координат вздовж вектора нормалі $\vec{\omega}$.

Тренувальні дані можуть бути лінійно роздільними. Тоді ми можемо обрати дві паралельні гіперплощини, які розділяють два класи даних таким чином, що відстань між ними є якомога більшою. Утвориться область, обмежена гіперплощинами. Її ми називаємо роздільною (англ. margin), а максимально розділова гіперплощина є гіперплощиною, яка лежить посередині між двома роздільними площинами кожного з класів. Ці гіперплощини може бути описані рівняннями:

$\vec{\omega} \cdot \vec{x} - b = 1$ та $\vec{\omega} \cdot \vec{x} - b = -1$

Приклад роботи класифікатора зображено на Рисунку 1.4

.

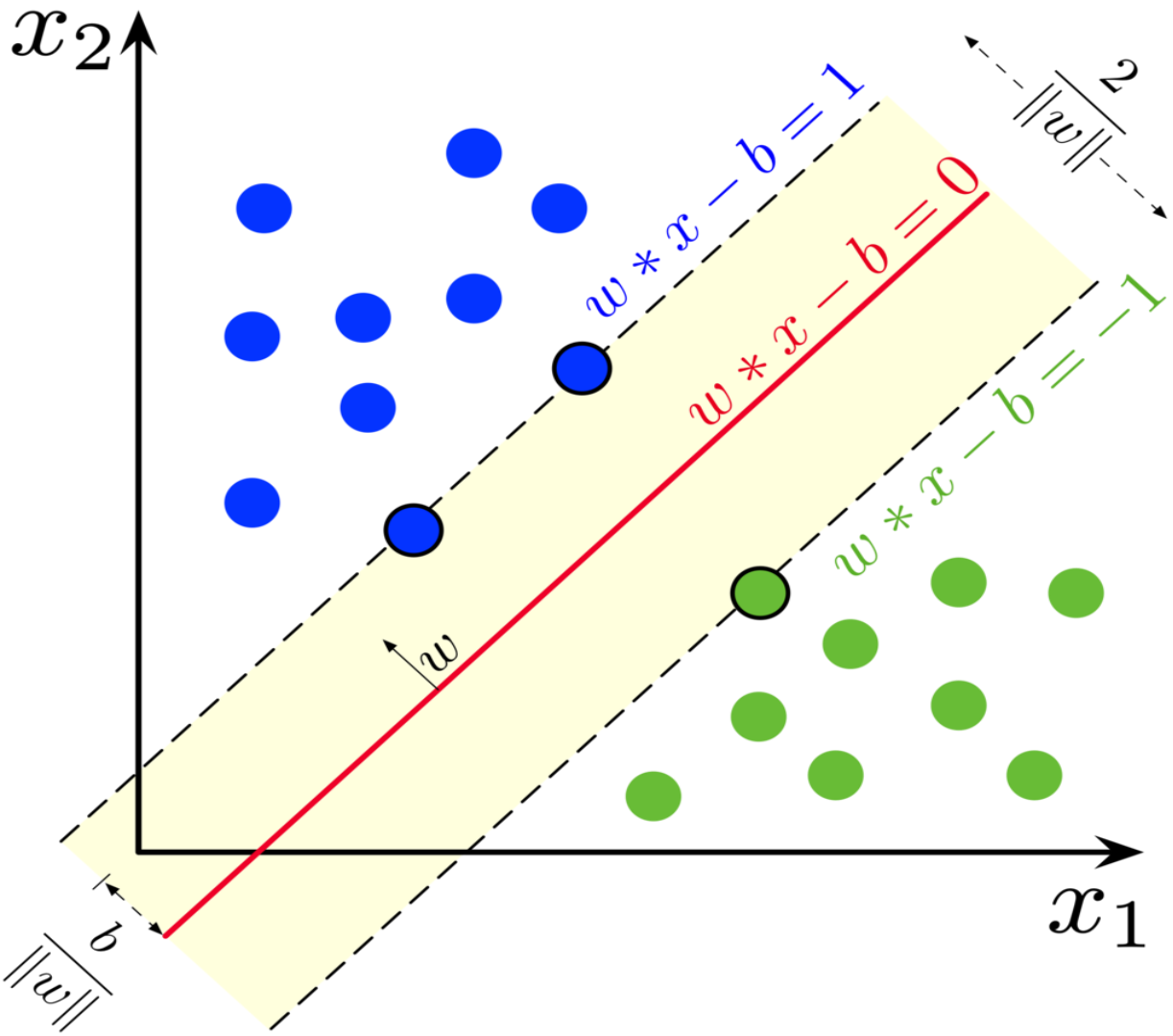

Рисунок 1.4- Візуалізація класифікатора

Максимально розділова гіперплощина та межі для SVM, натренованої зразками з двох класів. Зразки на межах називаються опорними векторами[9].

1.5 Наївний баєсів класифікатор

У машинному навчанні наївні класи Бейєса є сімейством простих імовірнісних класифікаторів, заснованих на застосуванні теореми Байєса з сильними (наївними) припущеннями про незалежність між функціями. Наївний Байєс широко вивчався з 1950-х років. Він був введений під іншим ім'ям в текстовий пошук для спільноти на початку 1960-х років і залишається популярним (базовим) методом для класифікації тексту, вирішує проблему розгляду документів як однієї категорії

.

(наприклад, спам, новини, спорт або політика тощо), тобто класифікації з частотою слів як функції. За умови відповідної попередньої обробки, цей метод є конкурентоспроможним у цьому застосуванні за допомогою більш просунутих методів, включаючи метод опорних векторів. Він також знаходить застосування в автоматичній медичній діагностиці. Класифікатори наївного Байєса мають високу масштабованість, що потребує ряду параметрів, лінійних за кількістю змінних (функцій/предикторів) у проблемі навчання. Тренування максимальної ймовірності можна зробити, оцінюючи вираз із замкнутою формою, який приймає лінійний час, а не дорогим ітераційним наближенням, як це використовується для багатьох інших типів класифікаторів. У статистиці та комп'ютерній літературі наївні моделі Байєса відомі під різними назвами, в тому числі простими Байєсами та незалежними Байєсами. Всі ці імена вказують на використання теореми Байєса в правилі рішення класифікатора, але наївний Байес не є (обов'язково) байєсівський методом.

Для деяких типів моделей вірогідності наївні класифікатори Байєса можуть бути дуже ефективними. У багатьох практичних додатках оцінка параметрів для наївних моделей Байеса використовує метод максимального правдоподібності; Іншими словами, можна працювати з наївною моделлю Байєса, не приймаючи вірогідність Байєса або використовуючи будь-які байєсівські методи. Незважаючи на наївний дизайн і, мабуть, надто спрощені припущення, наївні класифікатори Байєса працювали досить добре в багатьох складних реальних ситуаціях. У 2004 році аналіз проблеми класифікації Байєса показав, що існують обгрунтовані теоретичні причини для очевидної неімовірнісної ефективності наївних класифікаторів Байєса. Проте всеосяжне порівняння з іншими алгоритмами класифікації в 2006 році показало, що класифікація Байєса випереджає інші підходи, такі як дерева рішень або випадкові ліси. Перевага наївного Байеса полягає в тому, що потрібно лише невелику кількість навчальних даних для оцінки параметрів, необхідних для класифікації.

.

Модель наївного байєсівського класифікатора

Імовірнісна модель для класифікатора - це умовна модель: $p(C|F_1, \dots, F_n)$. Над залежною змінною класу C з малою кількістю результатів або класів, залежна від кількох змінних $F_1, \dots, F_n$. Проблема полягає в тому, що коли кількість властивостей n дуже велике або коли властивість може приймати велику кількість значень, тоді будувати таку модель на імовірнісних таблицях стає неможливо. Тому ми формулюємо модель, щоб зробити її легко піддається обробці. Використовуючи теорему Байеса запишемо:

$$p(C \mid F_1, \dots, F_n) = \frac{p(C)\, p(F_1, \dots, F_n \mid C)}{p(F_1, \dots, F_n)}$$

На практиці цікавий лише чисельник цього дробу, так як знаменник не залежить від C і значення властивостей F_i дані, тому знаменник - константа. Чисельник еквівалентний спільній ймовірнісній моделі : $p(\mathrm{C}, F_1, \dots, F_n)$. Спільна імовірнісна модель може бути виражена як[11]:

$$p(C, F_1, \dots, F_n) = p(C)\, p(F_1 \mid C)\, p(F_2 \mid C)\, p(F_3 \mid C) \cdot \dots \cdot p(F_n \mid C) =$$

$$= p(C) \prod_{i=1}^{n} p(F_i \mid C).$$

1.6 Дерево ухвалення рішень

Дерево рішень - це інструмент підтримки прийняття рішень, який використовує деревоподібну модель рішень та їх можливі наслідки, включаючи випадкові результати, витрати ресурсів та корисність. Це один з способів відображення алгоритму, який містить лише твердження з умовним керуванням.
Дерева рішень зазвичай використовуються в дослідженнях операцій, зокрема в аналізі рішень, щоб допомогти визначити стратегію, яка, найімовірніше, досягне мети, але також є популярним інструментом машинного навчання.

Дерево рішень є схемою, подібною до структури, в якій кожен внутрішній вузол представляє "тест" на атрибуті (наприклад, при підкиданні монети випав орел чи решка), кожна гілка являє собою результат тесту, а кожен вузол

.

представляє собою розділення гілки (рішення приймається після обчислення всіх атрибутів). Шляхи від кореня до листа являють собою правила класифікації.

У процесі аналізу рішень дерево рішень тісно пов'язане з діаграмою впливу і використовується як візуальний та аналітичний інструмент підтримки прийняття рішень, де розраховуються очікувані значення (або очікувана корисність) конкуруючих альтернатив.

Дерево рішень складається з трьох типів вузлів:

- Рішення вузлів - як правило, представлені квадратами
- Шаблонні вузли - зазвичай представлені колами
- Кінцеві вузли - зазвичай представлені трикутниками

Рішення дерев зазвичай використовуються в операціях дослідження та управління операціями. Якщо на практиці рішення повинні бути прийняті в Інтернеті без відкликання за додатковими заннями, то дерево рішень має бути представлено як алгоритм моделювання в режимі он-лайн вибору. Інше використання дерев рішень є як описовий засіб для обчислення умовних ймовірностей.

Приклад дерева рішень зображено на Рисунку 1.5.

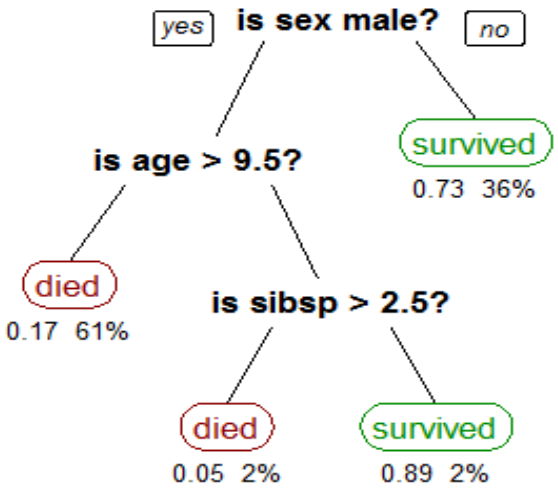

Рисунок 1.5- Візуалізація класифікатора

Типологія дерев

Дерева рішень, які використовуються в обробці даних, бувають двох основних типів:

- Дерево для класифікації, коли результат, що передбачається є класом, до якого належать дані;
- Дерево для регресії, коли результат, що передбачається можна розглядати як дійсне число (наприклад, ціна на будинок, або тривалість перебування пацієнта в лікарні).

Згадані вище терміни вперше були введені Брейманом. Перераховані типи мають деякі подібності (рекурсивний алгоритми побудови), а також деякі відмінності, такі, як критерії вибору розбиття в кожному вузлі. Деякі методи дозволяють побудувати більш одного дерева рішень (ансамблі дерев рішень):

- Беггінг над деревами рішень, найбільш ранній підхід. Будує кілька дерев рішень, неодноразово інтерполює дані з заміною (бутстреп), і в якості консенсусного відповіді видає результат голосування дерев (їх середній прогноз);
- Класифікатор «Випадковий ліс» заснований на беггінзі, проте як додаток до нього випадковим чином вибирає підмножину ознак в кожному вузлі, з метою зробити дерева більш незалежними;
- Бустінг над деревами може бути використаний для задач як регресії, так і класифікації. Одна з реалізацій бустінга над деревами, алгоритм XGBoost, неодноразово використовувався переможцями змагань з аналізу даних.
- «Обертання лісу» - дерева, в яких кожне дерево рішень аналізується першим застосуванням методу головних компонент (PCA) на випадкові підмножини вхідних функцій

Алгоритми побудови дерева

Ми можемо описати загальну схему побудови дерев рішень, виділивши наступні аспекти:

- Обираємо якийсь атрибут Q, він стає нашим коренем дерева.

.

- Якщо атрибут Q має і значень, то :
    - В дереві ми залишаємо ті дані, у яких значення атрибута Q дорівнює і
    - Застосовуємо рекурсію і будуємо гілки головного дерева

Є різні способи вибирати черговий атрибут:

- Алгоритм ID3, де вибір атрибута відбувається на підставі ентропії, або на підставі критерію Джині.
- Алгоритм C4.5 (поліпшена версія ID3), де вибір атрибута відбувається на підставі нормалізованого приросту інформації (англ. Gain Ratio)
- Алгоритм регресійних та класифікаційних дерев CART і їх модифікації різні модифікації.
- Автоматичний детектор взаємодії Хі-квадрат (CHAID). Виконує багаторівневе поділ при розрахунку класифікації дерев;
- MARS: розширює дерева рішень для поліпшення обробки цифрових даних.

На практиці в результаті роботи цих алгоритмів часто виходять занадто деталізовані дерева, які при їх подальшому застосуванні дають багато помилок. Це пов'язано з явищем перенавчання. Для скорочення дерев використовується відсікання гілок (англ. Pruning)[12].

1.7 Алгоритм Random Forest

Випадкові ліси (Random Forest) або ліси випадкових рішень - це метод вивчення ансамблю для класифікації, регресії та інших завдань, які працюють шляхом побудови безлічі дерев рішень під час навчання та виведення класу, який є одним із класів (класифікація) або середнім прогнозуванням (регресією) окремих дерев. Рішення дерев випадкового лісу вирішують проблему перенавчання окремих дерев рішень на навчальній вибірці. Перший алгоритм Random Forest був створений Тін Кам Хо з використанням методу випадкового підпростору, який, за формулюванням Хо, є способом реалізації підходу "стохастичної дискримінації" до класифікації, запропонованої Євгеном Клейнбергом.

.

Розширення алгоритму було розроблено Лео Брейман та Адель Катлер, а "Випадкові ліси" (Random Forest) - це їх торгова марка. Розширення поєднує в собі концепцію "мішків" Бреймана і випадковий вибір функцій, представлених спочатку Хо, а пізніше самостійно Амітом і Джеманом для побудови колекції дерев вирішення з контрольованою дисперсією.

Рішення дерев є популярним методом для різних завдань машинного навчання. Вузьке вивчення дерева "наближається до відповідності вимогам, щоб виступати як процедура незавершеного виявлення даних", - стверджує Хасти та співавтори, - "оскільки вона є інваріантною при масштабуванні та інших різноманітних перетвореннях значень ознак, є надійною до включення невідповідних функцій, а також створює моделі для перевірки, однак вони рідко точні."

Зокрема, дерева, які мають велику глибину, схильні до вивчення дуже нерегулярних моделей: вони накладають набори тренувань, тобто мають низький рівень зсуву, але дуже високу дисперсію. Випадкові ліси - це спосіб усереднення кількох дерев рішень, які тренуються на різних ділянках одного і того ж навчального комплексу, з метою зменшення дисперсії. Це відбувається за рахунок невеликого збільшення упередженості та деякої втрати інтерпретації, але в цілому значно підвищує продуктивність у фінальній моделі.

Оцінка важливості змінних

Випадкові ліси можуть бути використані для класифікації важливості змінних у регресії чи класифікації природним шляхом. Наступна техніка була описана в оригінальному документі Бреймана і реалізована в пакеті R randomForest.

Першим кроком у вимірюванні значення змінної в наборі даних є відповідність випадкового лісу для даних. Під час процесу створення випадкового лісу помилка для кожної точки даних фіксується та усереднюється за лісом (помилки на незалежному тестовому наборі можуть бути замінені, якщо пакетування не використовується під час тренувань).

.

Для вимірювання важливості j-ї функції після тренування значення j-ї функції змінюються серед навчальних даних, а помилка знову підраховується на цьому наборі даних. Оцінка важливості для j-ї функції обчислюється шляхом усереднення різниці в помилках до і після перестановки над усіма деревами. Оцінка нормується стандартним відхиленням цих відмінностей.

Особливості, які дають великі значення для цієї оцінки, класифікуються як більш важливі, ніж функції, які дають невеликі значення. Статистичне визначення змінного значення було дано та проаналізовано Роуцином Чжу. Цей метод визначення змінного значення має деякі недоліки. Для даних, включаючи категоріальні змінні з різною кількістю рівнів, випадкові ліси з ухилом використовуються для цих атрибутів з більшою кількістю рівнів. Для вирішення проблеми можна використати такі методи, як часткові перестановки та зростаючі об'єднані дерева. Якщо дані містять групи поєднаних функцій, подібних до результатів, то менші групи віддають перевагу більшим групам.

Переваги і недоліки

Серед інших методів пошуку даних дерева рішень мають різні переваги:

- Просто зрозуміти та інтерпретувати.
- Здатний обробляти як цифрові, так і категоричні дані.
- Не вимагає підготовки даних.
- Використовує модель білої коробки.
- Можна перевірити модель, використовуючи статистичні тести.
- Добре працює з великими наборами даних.
- Відображає людське прийняття рішень більш тісно, ніж інші підходи.
- Дерева рішень можуть наближати будь-яку булеву функцію екв. XOR

До недоліків можна віднести:

- Дерева можуть бути дуже нестійкими. Невелика зміна навчальних даних може призвести до значних змін у дереві та, як наслідок, до остаточних прогнозів

.

- Легко створювати надскладні дерева, які в узагальненому вигляді підходять під навчальні дані, але погано прогнозують тестові дані [13].

1.8 Перевірка результатів роботи алгоритмів

Постановка задачі: Потрібно класифікувати відгуки користувачів ресторанів за двома категоріями – позитивними та негативними.

Збір даних: Для вирішення задачі класифікації була обрана готова вибірка даних з сайту Super Data Science [5]. В цій вибірці позитивні відгуки класифіковані як ‘1’, а негативні як ‘0’.

Результати розв’язку задачі: На прикладі вибірки з 1000 розмічених відгуків про ресторан (позитивних або негативних) були проаналізовані результати роботи 6 методів машинного навчання: логістична регресія, SVM, KNN, Наївний Байес, дерева ухвалення рішень та алгоритм Random Forest. Для навчання алгоритмів та побудови моделей була сформована вибірка з 800 відгуків, з 200 відгуків була сформована тестова вибірка, на якій перевірялась точність класифікації. За результатами класифікації були сформовані матриці помилок для кожного алгоритму, за якими були визначені наступні показники точності (Точність = вірно класифіковані думки/ всі думки):

1) Логістична регресія – 71%
2) KNN – 61%
3) SVM -72%
4) Наївний Байес – 73%
5) Дерево рішень – 71%
6) Random Forest -72%

Показник детальності і повноти (Детальність = це частка знайдених думок, які мають відповідні до запиту; Повнота = це частка релевантних думок, які успішно знайдені системою пошуку відносно загальної кількості релевантних думок)

1) Логістична регресія: Детальність – 78%, Повнота – 67%

.

2) KNN: Детальність – 76%, Повнота – 57%
3) SVM : Детальність – 76%, Повнота – 69%
4) Наївний Байес : Детальність – 57%, Повнота – 82%
5) Дерево рішень : Детальність – 76%, Повнота – 68%
6) Random Forest : Детальність – 90%, Повнота – 65%

Для всіх алгоритмів обиралися параметри за замовчуванням аби перевірити їх точність у порівнянні з іншими алгоритмами в однакових умовах. У підсумку, найкращими алгоритмами для обробки природніх мов виявилися алгоритми Наївний Байес, SVM та Random Forest.

Висновки до розділу 1

В даному розділі магістерської дисертації були розглянуті алгоритми машинного навчання, їх математичне обгрунтування, для вирішення задачі класифікації, а саме: логістична регресія, алгоритм к-найближчих сусідів, метод опорних векторів (SVM), наївний баєсів класифікатор, дерево ухвалення рішень і алгоритм Random Forest . За допомогою цих алгоритмів були отримані практичні результати їх роботи на вибірці із 1000 позитивних та негативних відгуків щодо ресторану, проведений порівняльний аналіз.

.

2 МАТЕМАТИЧНА МОДЕЛЬ “МІШОК СЛІВ” (“Bag of words”)

Математичний метод завжди слід двом принципам:

1) *Узагальнення (абстрагування).* Об'єкти вивчення в математиці - це спеціальні суті, які існують тільки в математиці і призначені для вивчення математиками. Математичні об'єкти утворюються шляхом узагальнення реальних об'єктів. Вивчаючи який-небудь об'єкт, математик зауважує тільки деякі його властивості, а від інших відсторонюється. Так, абстрактний математичний об'єкт «число» може в реальності позначати кількість гусей в ставку або кількість молекул в краплині води.
2) *Строгість міркувань*. В математиці результати не перевіряються експериментальним шляхом, але вони доводяться, бо підпорядковуються певним правилам міркуваннями (доказам), які служать єдиним способом обгрунтування вірності того чи іншого твердження.

Щоб вивчати за допомогою математичних методів лінгвістичні об'єкти (мови, тексти), необхідно виділити з об'єкта його властивості, які видаються важливими для вивчення та строго визначити ці властивості.

Отримана таким чином абстракція буде математичною моделлю реального об'єкта (формальною мовою, векторної моделлю тексту та інше).

2.1 Модель “мішок слів”

Найпростіша модель тексту "мішок слів" (bag-of-words) являє собою сумативну єдність (не систему) складових текст слів.

Основний об'єкт моделі мішка слів - це слово, забезпечене єдиним атрибутом, частотою розповсюдженості цього слова. У моделі тексту "мішок слів" враховується тільки кількість входжень конкретних слів в початковому тексті, при цьому ігноруються: порядок слів у документі,

морфологічні форми подання слів.

.

Модель тексту "мішок слів" була запропонована в 1975 році Дж. Солтоном, і в даний час є однією з найбільш поширених в самих різних областях лінгвістичних досліджень і сервісів, як правило, в якості основи для більш складних, перш за все "векторних моделей тесту".

Мішок слів використовується в машинному навчанні на основі текстів в якості одного з основних об'єктів вивчення.

2.2 Модель “мішок термів”

Корисним узагальненням формальної моделі тексту "мішок слів" (bag-of-words) може служити модель "мішок термів" (bag-of-terms).

Основний об'єкт моделі мішка термів - це терм, забезпечене єдиним атрибутом, частотою появи терма в тексті.

Терм - символьний вираз об'єкта формальної моделі (мови, системи)

Терм (формальне визначення): символьний вираз: t (X1, X2, ..., Xn), де t - ім'я терма, так званий функтор або «функціональна буква», а X1, X2, ..., Xn - терми, структуровані або найпростіші.

Терми можуть містити вільні змінні (параметри), фіксація значень яких однозначно визначає відповідно до семантичними правилами мови деякий об'єкт - значення Терма при даних значеннях його вільних змінних.

У логіко-математичному обчисленні терм є аналогом іменника або прикметника природної мови.

У моделі bag-of-terms, у якості термів можуть бути розглянуті будь-які символьні вирази тексту, в тому числі - розділові знаки.

2.3 Частотна модель тексту

Для кожного слова з набору моделі "мішок слів" може вказується певну «вагу». Таким чином, модель тексту представляє собою безліч пар «слово - вага». При цьому ваги можуть присвоюватися словами або основам слів.

.

Методи визначення ваги слів:

1) *Бінарний метод (поширене позначення - BI, від binary)*. Визначається тільки наявність або відсутність деяких термінів в документі. Застосовується для логічного інформаційного пошуку і автоматичної рубрикації текстів методами нейронних мережевих класифікаторів ART і SOM.
2) *Кількість входжень (слів в документ)*. Передбачає невідповідність оцінки для текстів різної довжини - більшу вагу отримуватимуть більш об'ємні тексти, так як в них більше слів;
3) *Частота входження слова в документі (TF - term frequency)*. Частота обчислюється як відношення числа входження слова до загальної кількості слів тексту. При відносній простоті ця характеристика забезпечує прийнятний результат для методів інформаційного пошуку і класифікації (передбачає невідповідність оцінки для текстів різної довжини - недооцінюються довгі документи, так як в них більше слів і середня частота слів в тексті нижче).
4) *Логарифм частоти входження слова (LOGTF)*. Вага входження слова в текст документа визначається як 1 + log (TF), де TF - частота терміна. Використання логарифмічною шкали дозволяє зробити модель більш стійкою до переоцінки текстів різного обсягу.
5) *Зворотній частота документів (IDF - inverse document frequency)*. Параметр є інверсією частоти, з якою зустрічається термін в документах.

2.4 Векторна модель тексту

Векторна модель (vector space model) - представлення текстів векторами з одного спільного для всієї колекції текстів векторного простору.

Векторна модель є основою вирішення завдань інформаційного пошуку:

це включає класифікація та попередню кластеризацію документів, пошук документа за запитом і інші функції.

.

Текст (документ) у векторній моделі розглядається як неврегульована множина термів. Терми - слова, з яких складається текст, а також інші значимі в моделі елементи тексту (числа, знаки пунктуації, спеціальні позначення, акроніми тощо).

Вектор, який є модельним поданням тексту в векторному просторі, утворюється упорядкуванням ваг всіх термів (включаючи ті, яких немає в конкретному тексті). Розмірність цього вектора і розмірність простору вектора, є аналогічною до кількості різних термів у всій колекції. Ця кількість є однаковою і для всіх текстів (документів) колекції.

Формально можна визначити:
dj = (w1j, w2j, ..., wnj), де dj - векторне подання j-го тексту, wij - вага i-го терма в j-м документі, n - загальна кількість різних термів у всіх документах колекції.

Таке представлення для всіх текстів дозволяє застосовувати метод для машинного навчання бо, можна, наприклад, знаходити евклідову відстань між точками простору. Це дозволить вирішити задачу подібності текстів – якщо точки будуть згруповані близько одна від одної, то можна зробити висновок, що тексти схожі. Пошуковий запит представляється як вектор того ж простору, якщо потрібно провести пошук за текстами і за аналогічним методом можна обчислити відповідність документів запиту.
Для повного визначення векторної моделі необхідно вказати, яким саме чином буде знаходитись вага терма в документі.

2.5 Частотний словник як векторна модель тексту

Найпростішою векторною моделлю тексту (vector space model) може служити побудований на основі моделі "мішок слів" частотний словник тексту.
Частотний словник може бути інтерпретований як найпростіша векторна модель тексту (елементи - кортежі зі слів і параметра їх частотності).

.

Частотний аналізатор визначає для кожного слова vi з словникового простору (системного словника) V його частоту входження fi ≥ 0 в даний текст t = t1t2t3 ... tk. Частотну характеристику f можна розглядати як точку в просторі ознак F, відповідну тексту t. Частотна характеристика - це вектор векторного простору F.

Довжина f = (f1, ... fn) дорівнює кількості слів в словнику V, кожна компонента fi це ціле невід'ємне число. Таким чином, на вході маємо текст t і словник V, на виході крапку в просторі ознак F. Системний словник V може бути заданий "неявно" як сукупність всіх слів тексту, який моделюється.

2.6 Латентно-семантичний аналіз

Модель "мішок слів" використовується в латентно-семантичному аналізі (ЛСА, LSA). Латентно-семантичний аналіз (ЛСА, LSA) - метод обробки інформації на природній мові, який встановлює взаємозв'язок між текстами (документами) і термінами, що в них зустрічаються.

LSA використовується для вилучення контекстно-залежних значень лексичних одиниць на основі факторного аналізу та статистичної обробки великих корпусів текстів.

LSA запатентований в 1988 р. (Скотт Дервестер, Сюзан Дюмейнс, Джорж Фурнас, Річард Харшман, Томас Ландоер, Карен Лохбаум і Лін Стрітер).
LSA вперше застосований для отримання псевдодокументів, автоматичної індексації документів в мережі, виявлення семантичної структури тексту.

LSA застосовують для:

- пошуку інформації (індексація документів) - області інформаційного пошуку даний підхід називають латентно-семантичним індексуванням (ЛСІ);
- класифікації документів; уявлення баз знань;
- побудови когнітивних моделей (моделей розуміння).

LSA представимо (моделюємо) тришарової нейромережею: безліч слів (термів);

.

безліч документів, які відповідають певним ситуаціям; (Латентний шар) безліч вузлів з різними ваговими коефіцієнтами, що пов'язують перший і другий шари.

В якості вихідної інформації LSA використовує матрицю терми-на-документи, що описує тренувальний набір даних для машинного навчання. Елементи отриманої матриці, яка подається алгоритмам, містять ваги. Ці ваги враховують участь терма в усіх документах (TF-IDF) і частоти використання кожного терма в кожному документі.

Найбільш поширений варіант LSA заснований на використанні розкладання діагональної матриці за сингулярними значеннями (SVD - Singular Value Decomposition). За допомогою SVD-розкладання будь-яка матриця розкладається в безліч ортогональних матриць, лінійна комбінація яких є досить точним наближенням до початкової матриць.

LSA зазвичай передує підготовчими операціями:

- виняток стоп-символів (стоп-слів, стоп-термів).
- нормалізація тексту
- виняток малочастотних умов (зустрічаються в єдиному екземплярі). Ця операція сильно спрощує математичні обчислення.

2.7 Нормалізація

Нормалізація - видалення з вихідного тексту граматичної інформації (відмінки, числа, дієслівні види і часи, рід слів і так далі).

Два інших алгоритму - стемінг і лематизації - намагаються досягти такого ж ефекту, але глибина перетворення тексту в них менше.

Основна проблема, що виникає при використанні стеммера - це обробка слів, які при утворенні різних граматичних форм змінюють не тільки закінчення, а й основу слова.

.

Нормалізація тексту не використовує стемінг, тому вона позбавлена недоліків втрати релевантності через особливості української чи російської зміни слова. Лемматізатор за своїми результатами знаходиться набагато ближче до нормалізатора. Однак він застосовує спрощений аналіз слів, не враховуючи контекст. Це призводить до неоднозначності при визначенні частини мови.
Нормалізація вимагає обов’язкового морфологічного аналізу, що розпізнає частини мови з урахуванням контексту і численних правил узгодження (без нього нормалізація буде давати значну кількість помилкових результатів).

2.8 Стемінг

Стемінг - процес знаходження основи слова для заданого вихідного слова (є частиною процесу нормалізації тексту).
Основа слова необов'язково збігається з морфологічним коренем слова.
В області комп'ютерних наук перша публікація по стемінгу датується 1968 роком.

Стемінг застосовується в пошукових системах для розширення пошукового запиту користувача. Конкретний спосіб вирішення завдання пошуку основи слів називається алгоритмом стемінгу, а конкретну реалізацію - Стемер.

Стемінг важливий:

- при невеликому наборі текстів
- для української (російської) мови (як флективної).

Українська мова належить до групи флективних синтетичних мов, тобто мов, в яких переважає словотвір з використанням афіксів, що поєднують відразу кілька граматичних значень, тому дана мова допускає використання алгоритмів стемінгу. Українська мова має складну морфологічну змінність слів, яка є джерелом помилок при використанні стемінгу. Як вирішення проблеми можна використовувати поряд з класичними алгоритмами стемінгу алгоритми лематизації, які наводять слова до початкової базової формі.

.

Якщо тексти англійською мовою, то цей крок теж можна проігнорувати, тому що кількість варіацій тієї чи іншої словоформи в англійській мові значно менше ніж в українській чи російській.

Для стемінг популярний алгоритм Портера. Основна ідея стемера Портера полягає в тому, що існує обмежена кількість словоутворюючих суфіксів. Процес стемінгу слова не має включати використання коренів слів: використовується тільки база існуючих суфіксів, префіксів і словотворчі правила природної мови.

Алгоритм складається з п'яти кроків. На кожному кроці відсікається словоутворюючий суфікс і решта перевіряється на відповідність правилам (наприклад, для українських слів корінь слова повинен містити як мінімум одну голосну літеру чи звук). Якщо в кінці стемінгу отриманий корінь задовольняє встановленим правилам, то алгоритм переходить до наступного кроку або обирає наступне слово. Якщо це не так, то алгоритм вибирає інший префікс чи суфікс для відсікання [1].

Висновки до розділу 2

В даному розділі магістерської дисертації була розглянута математична модель “мішок слів”, “мішок термів”, частотна та векторна моделі тексту, поняття нормалізації тексту та стемінгу, принципи латентно-семантичного аналізу тексту. За допомогою представлення тексту у якості моделі “мішок слів” можна надалі застосувати один із алгоритмів машинного навчання і вирішити задачу аналізу тональності тексту.

.

3 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕНЬ ТА ПРОВЕДЕНИХ СОЦІОЛОГІЧНИХ ОПИТУВАНЬ

3.1 Моніторинг електоральних настроїв українців (Київський міжнародний інститут соціології)

КМІС, Центр Разумкова та Соціологічна група Рейтинг провели соціальне опитування в жовтні 2018 року. Результати та підхід описані до опитування описані нижче.

Вибірка дослідження: десять тисяч респондентів в усіх областях України (за виключенням населення тимчасово окупованих АР Крим та територій Донецької та Луганської областей). Показники, на основі яких формувалася вибірка: область проживання, тип поселення, стать та вік.

Статистична похибка: не більше одного відсотка.

Дослідження було проведено в жовтні та листопаді 2018 року.

За результатами цього дослідження отримані наступні результати:

- Ю.Тимошенко є лідером президентських передвиборчих перегонів. Вона випереджає найближчих конкурентів з різницею у два рази. Близько 20% виборців готові віддати за неї голос. Це ті виборці які визначилися за кого проголосують та мають намір взяти участь у волевиявленні. Володимир Зеленський, Петро Порошенко та Анатолій Гриценко мають на своєму боці близько 10% від загальної кількості виборців (Кожен). Іншу частину голосів, тобто менше 10% кожен) мають кандидати Юрій Бойко, Олег Ляшко, Святослав Вакарчук, Є.Мураєва. Рейтинг інших кандидатів – менше п’яти процентів. Статистика демонструє, що рейтинг Тимошенко відчутно зростає протягом 2018 року. Процент підтримки Святослава Вакарчука впав вдвічі, а рейтинг Володимира Зеленського навпаки зріс на цьому тлі.
- Кожен п’ятий громадянин вірить в те, що Юлія Тимошенко стане наступним президентом. Кожен восьмий громадян, який планує голосувати на виборах

.

вірять, що Петро Порошенко піде на другий президентський строк. Те, що переможе хтось з інших кандидатів вірять не більше 5% опитаних. Показник Петра Порошенка, з квітня по жовтень 2018, фактично не змінився. Натомість, кількість виборців, які вірять, що Юлія Тимошенко стане наступним президентом зросла вдвічі.

- Було проведено моделювання другого туру президентських виборів в усіх можливих електоральних парах. Юлія Тимошенко поки що перемагає в усіх таких парах. Натомість у другому турі Петро Порошенко має незначні шанси на перемогу, бодай в якійсь парі[3].

Більш розширені результати дослідження наведено на Рисунках №3.1-3.10 нижче:

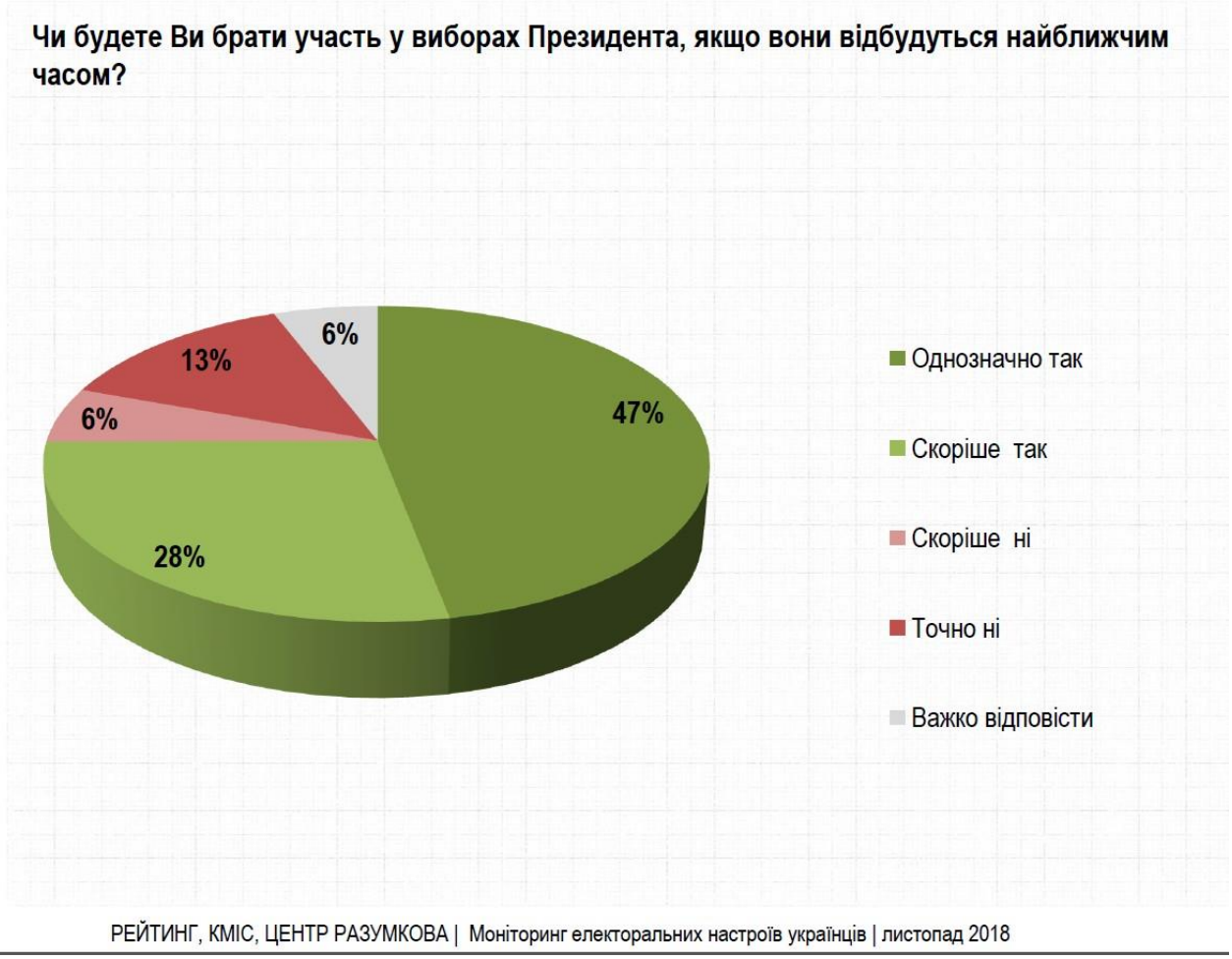

Рисунок 3.1- Результати дослідження

Якби вибори Президента України відбулися б наступної неділі, то як би Ви проголосували?

РЕЙТИНГ

	Серед усіх %	Серед тих, хто має намір голосувати, %	Серед тих, хто має намір голосувати і визначився, %
Тимошенко Юлія	12,7	16,0	20,7
Зеленський Володимир	7,6	8,8	11,4
Порошенко Петро	6,3	8,0	10,3
Гриценко Анатолій	6,2	7,7	9,9
Бойко Юрій	5,5	6,8	8,7
Ляшко Олег	4,9	5,9	7,6
Вакарчук Святослав	3,7	4,3	5,5
Мураєв Євгеній	3,2	3,9	5,1
Рабінович Вадим	2,3	2,8	3,7
Шевченко Олександр	2,1	2,6	3,3
Садовий Андрій	1,8	2,2	2,9
Безсмертний Роман	0,9	1,1	1,4
Наливайченко Валентин	0,9	1,1	1,4
Тарута Сергій	0,8	1,0	1,3
Кошулинський Руслан	0,5	0,6	0,7
Яценюк Арсеній	0,5	0,6	0,7
Добродомов Дмитро	0,4	0,5	0,6
Інший кандидат	3,3	3,6	4,7
Важко відповісти	21,5	22,6	
Не приймав би участі	15,1		

РЕЙТИНГ, КМІС, ЦЕНТР РАЗУМКОВА | Моніторинг електоральних настроїв українців | листопад 2018 25

Рисунок 3.2- Результати дослідження

Рейтинги кандидатів: ДИНАМІКА

РЕЙТИНГ

Серед тих, хто має намір голосувати і визначився з вибором, %

	вибори 2014	04'15	06'15	08'15	10'15	12'15	01'16	06'16	08'16	09'16	11'16	12'16	05'17	10'17	12'17	02'18	03'18	04'18	05'18	06'18	07'18	08'18	09'18	10'18
Ю. Тимошенко	13	11	13	15	16	13	14	15	18	17	18	18	15	17	19	17	15	14	14	16	17	18	19	21
П. Порошенко	55	31	32	24	26	23	22	12	11	15	14	14	12	15	16	10	9	9	10	9	8	8	10	10
В. Зеленський																	6	8	9	9	9	8	11	11
А. Гриценко	5	6	8	6	6	6	6	7	8	6	8	8	10	10	10	10	11	11	11	12	10	11	10	10
Ю. Бойко		8	8	10	9	11	10	14	12	11	10	10	10	9	12	10	10	9	10	11	9	9	8	9
О. Ляшко	8	8	6	8	7	8	5	10	10	11	10	9	10	8	9	8	6	8	8	9	8	9	7	8
С. Вакарчук																7	8	9	8	9	7	8	7	6
Є. Мураєв																					4	–	–	5
В. Рабінович	2									8	7	7	10	10	8	8	8	6	7	8	4	6	8	4
О. Шевченко																							3	3
А. Садовий		6	7	9	12	11	12	10	9	6	7	8	8	7	6	6	6	3	3	3	2	2	4	3
О. Тягнибок	1	3	3	3	4	6	6	3	4	5	5	5	5	4	4	3	5	2	2	3	3	2	2	–
В. Наливайченко																	1	2	2	2	1	2	2	1
Інший кандидат	16	27	23	25	20	22	25	29	28	21	21	21	20	20	17	21	17	17	16	11	18	17	9	9

Рисунок 3.3- Результати дослідження

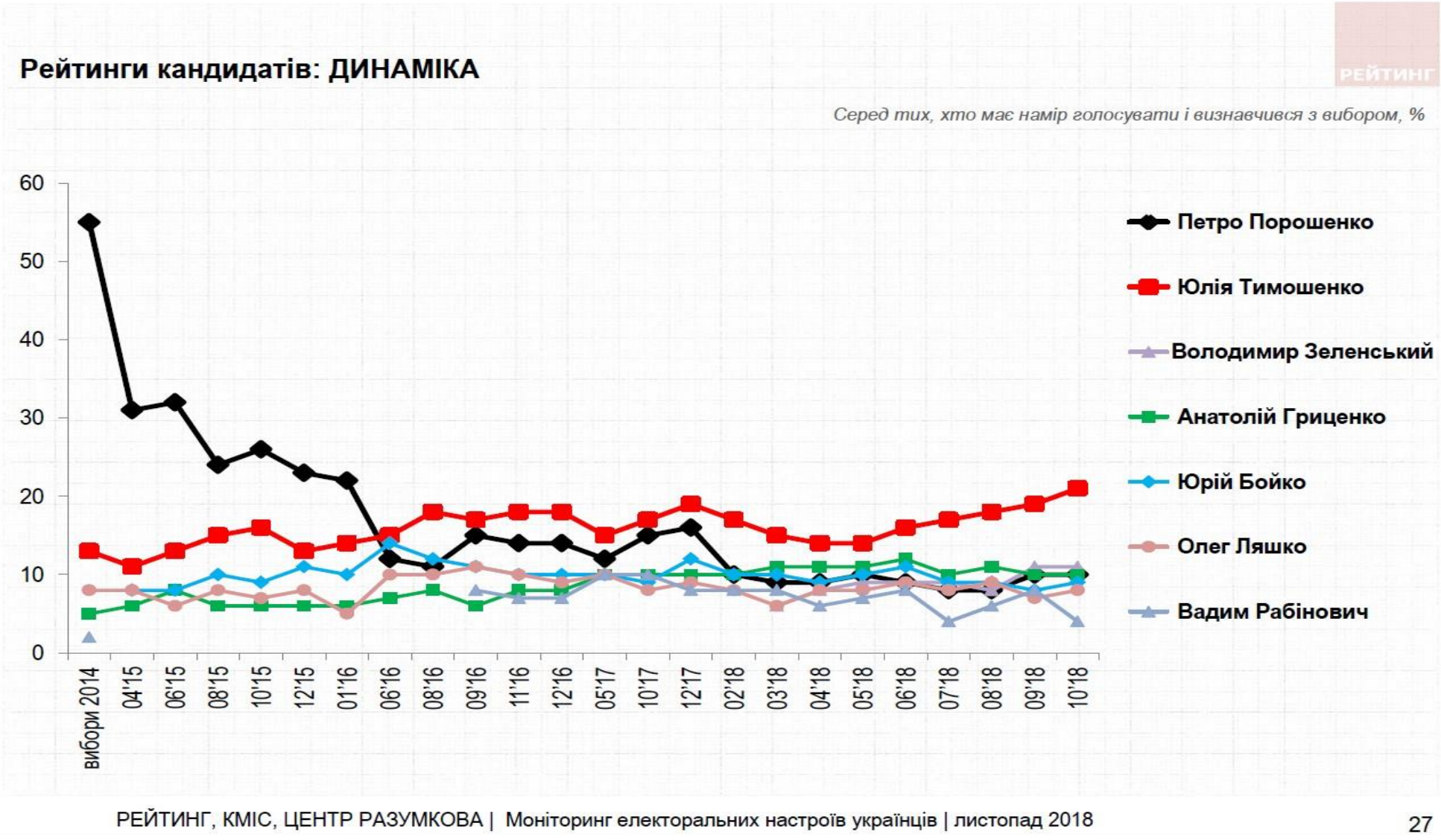

Рисунок 3.4- Результати дослідження

Рейтинг кандидатів: регіони

РЕЙТИНГ

% від усіх респондентів

	Захід	Галичина	Центр	Північ	Київ	Південь	Схід	Донбас
Тимошенко Юлія	14	12	18	17	11	10	10	8
Зеленський Володимир	7	5	8	8	7	10	9	6
Порошенко Петро	6	9	7	8	11	4	4	4
Гриценко Анатолій	7	12	7	7	6	3	4	4
Бойко Юрій	3		3	4	4	8	9	12
Ляшко Олег	7	4	6	6	3	4	4	5
Вакарчук Святослав	4	8	4	3	5	2	2	3
Мураєв Євгеній	1		2	2	3	5	7	6
Рабінович Вадим	1		1	1	1	4	3	6
Шевченко Олександр	4	7	2	2		1	1	
Садовий Андрій	2	5	2	1	1	1	1	1
Наливайченко Валентин	1	1	1	1	3		1	
Безсмертний Роман	1	2	1	1	1	1		1
Тарута Сергій				1	1	1	1	2
Кошулинський Руслан	1	2						
Яценюк Арсеній		1		1		1		
Добродомов Дмитро	1	1	1	1	0	0	0	0
Інший кандидат	3	3	2	3	4	5	4	3
Не приймав би участі	13	9	13	13	17	20	18	19
Важко відповісти	24	19	23	21	22	18	23	21

РЕЙТИНГ, КМІС, ЦЕНТР РАЗУМКОВА | Моніторинг електоральних настроїв українців | листопад 2018 28

Рисунок 3.5- Результати дослідження

А за кого з кандидатів Ви не проголосуєте за жодних обставин?

РЕЙТИНГ

	Серед усіх %	Серед тих, хто має намір голосувати, %
Порошенко Петро	50,2	51,4
Яценюк Арсеній	27,2	27,6
Тимошенко Юлія	27,3	27,5
Ляшко Олег	24,4	25,5
Бойко Юрій	20,3	22,0
Рабінович Вадим	17,2	18,3
Мураєв Євгеній	10,1	10,1
Зеленський Володимир	8,1	8,4
Садовий Андрій	8,4	8,1
Тарута Сергій	8,0	7,8
Вакарчук Святослав	7,5	7,7
Гриценко Анатолій	7,1	6,9
Наливайченко Валентин	6,6	6,3
Кошулинський Руслан	5,3	5,2
Шевченко Олександр	5,6	5,2
Безсмертний Роман	5,4	4,9
Добродомов Дмитро	4,9	4,7
Інший кандидат	2,3	1,7
Важко відповісти	19,6	14,6

РЕЙТИНГ, КМІС, ЦЕНТР РАЗУМКОВА | Моніторинг електоральних настроїв українців | листопад 2018 29

Рисунок 3.6- Результати дослідження

А хто, на Вашу думку, стане наступним Президентом України?

РЕЙТИНГ

	Серед усіх %	Серед тих, хто має намір голосувати, %
Тимошенко Юлія	17,6	20,1
Порошенко Петро	13,0	13,9
Зеленський Володимир	3,6	3,9
Гриценко Анатолій	2,3	2,9
Бойко Юрій	2,3	2,8
Ляшко Олег	2,2	2,6
Вакарчук Святослав	1,3	1,4
Рабінович Вадим	1,1	1,3
Мураєв Євгеній	1,0	1,2
Шевченко Олександр	0,9	1,0
Садовий Андрій	0,6	0,8
Тарута Сергій	0,4	0,4
Безсмертний Роман	0,3	0,4
Наливайченко Валентин	0,3	0,4
Кошулинський Руслан	0,2	0,3
Яценюк Арсеній	0,2	0,2
Добродомов Дмитро	0,1	0,1
Інший кандидат	2,7	2,7
Важко відповісти	50,0	43,8

РЕЙТИНГ, КМІС, ЦЕНТР РАЗУМКОВА | Моніторинг електоральних настроїв українців | листопад 2018 30

Рисунок 3.7- Результати дослідження

Рисунок 3.8- Результати дослідження

Рисунок 3.9- Результати дослідження

Рисунок 3.10- Результати дослідження

3.2 Соцопитування МРІ (Міжнародний республіканський інститут): передвиборчі настрої в Україні

Наступні вибори президента України заплановано провести 31 березня 2019 року. У зв'язку з цим, Центр аналізу та соціологічних досліджень Міжнародного республіканського інституту (МРІ) провів соціальне опитування серед виборців, щодо майбутнього переможця президентських виборів

Результати соціологічного опитування надають важливу інформацію щодо електоральних поглядів українців, їх думки щодо лідерів та аутсайдерів у президентських перегонах, проблеми передвиборчої кампанії, яка офіційно ще не стартувала.

МРІ наводить найголовніші показники із зібраної статистики. Повний звіт про результати соціологічного дослідження буде надано пізнише.

Рейтинги підтримки

Нижче наведено рейтинг кандидатів і який процент виборців готові за них проголосувати, якби вибори відбувалися у листопаді 2018 року (результати зображено на Рисунку 3.11):

.

Рисунок 3.11- Результати дослідження

Методологія

Соціологічна група «Рейтинг» провела опитування виборців на замовлення Міжнародного республіканського інституту. Опитування проводилося на всій території Україні (крім тимчасово окупованих територій Криму і Донбасу) з вересня по жовтень 2018 року. У якості метода опитування було обрано особисте інтерв'ю. В опитуванні прийняли участь близько трьох тисяч громадян України. Вікова група - старше 18 років, які мають можливість голосувати. Показники, на основі яких формувалася вибірка: область проживання, тип поселення, стать та вік. Коефіцієнт досяжності респондентів становить більше 60%, а допустима похибка не перевищує двох відсотків[18].

3.3 Результати соцопитування проведеного КМІС, Центром Разумкова та СОЦИС.

Нижче наведені результати соціологічного дослідження, проведеного КМІС, Центром Разумкова та СОЦИС.

Юлія Тимошенко має найбільший рейтинг і шанс на перемогу у президентських виборах. Найближчі 6 переслідувачів мають рейтинг близько 10% і приблизно однакові шанси на перемогу та проходження до другого туру виборів.

Кожен восьмий виборець, який вже визначився із голосом, готовий проголосувати за Юлію Тимошенко. Окрім неї до другого туру президентських виборів можуть потрапити Володимир Зеленський, Святослав Вакарчук, Юрій Бойко, Анатолій Гриценко, та чільний голова країни - Петро Порошенко. Їм віддати свій голос планують близько семи- восьми відсотків виборців, при цьому похибка дослідження складає один відсоток. Більше четверті опитаних виборців ще не визначилися із тим за кого будуть голосувати. Кожен восьмий виборець прогнозує перемогу для Юлії Тимошенко, 13,4% роблять ставку на чільного президента України . У інших кандидатів, на думку опитаних, немає шансів на перемогу.

.

Важливо відзначити, що близько 50% респондентів не мають уявлення хто переможе на виборах і відмовилися давати свій прогноз.

Дослідження тривало у серпні-вересні 2018 року. Інтерв'ю проведилися безпосередньо з виборцями за місцем постійного проживання респондентів. Загалом було опитано більше десяти тисяч осіб, віком від 18 років і старше.

Статистична похибка вибірки не перевищує одного відсотка [4].

Результати дослідження зображено на Рисунках № 3.12-3.16.

.

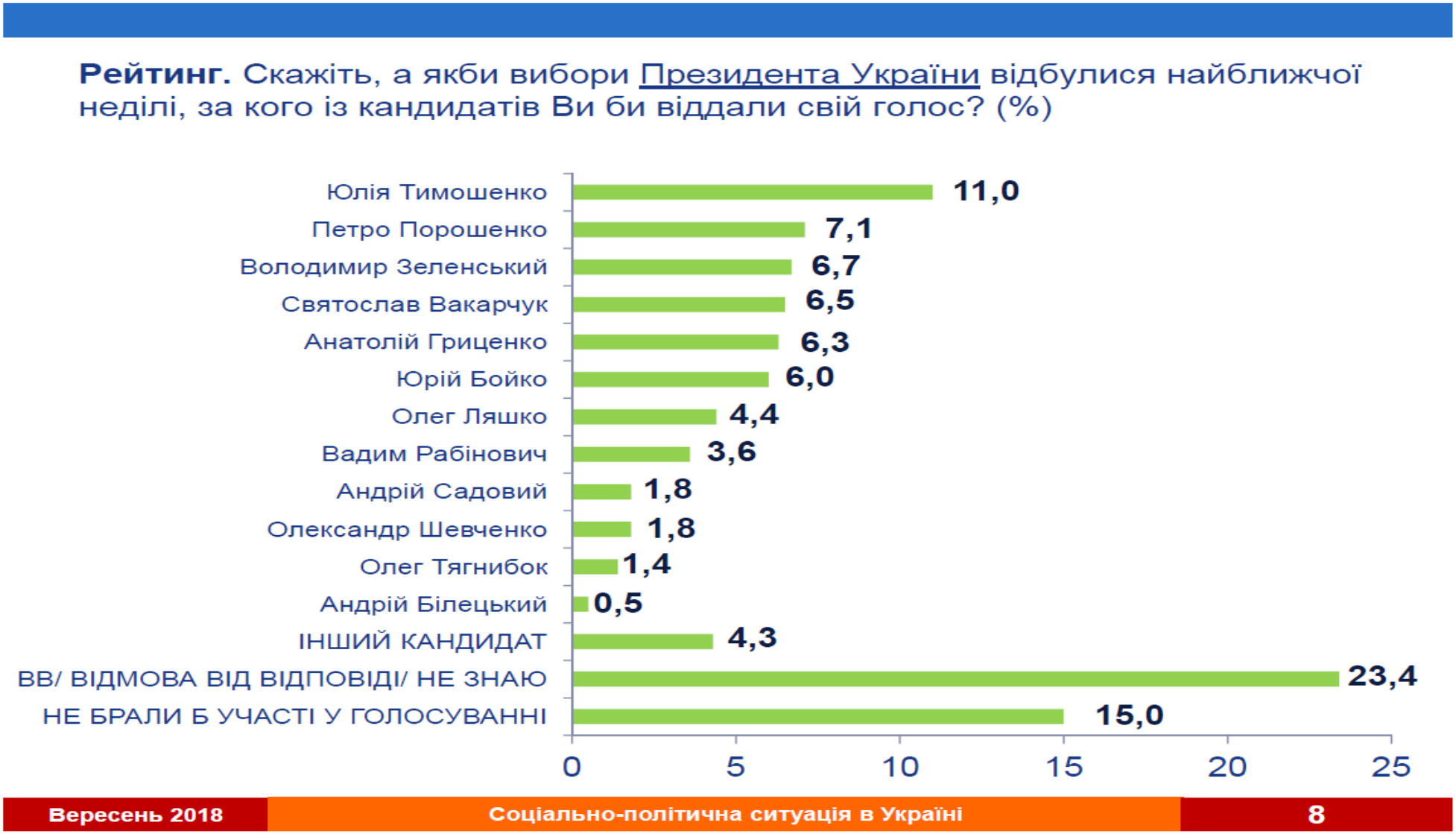

Рисунок 3.12- Результати дослідження

Рейтинг. Скажіть, а якби вибори Президента України відбулися найближчої неділі, за кого із кандидатів Ви би віддали свій голос? (%)

100% по кожному стовпчику	Україна загалом	Південь	Схід	Центр Сх	Центр Зх	Захід	Київ
Юлія Тимошенко	**11,0**	9,6	6,5	12,2	14,6	10,3	11,4
Петро Порошенко	**7,1**	4,0	4,2	6,1	8,3	9,7	11,0
Володимир Зеленський	**6,7**	9,8	6,6	7,3	5,8	4,3	8,3
Святослав Вакарчук	**6,5**	3,5	4,3	4,1	6,8	11,3	7,3
Анатолій Гриценко	**6,3**	4,5	3,0	5,5	6,7	9,2	8,6
Юрій Бойко	**6,0**	11,1	13,2	6,2	3,1	1,1	5,5
Олег Ляшко	**4,4**	3,8	3,2	7,2	5,3	2,9	2,3
Вадим Рабінович	**3,6**	6,8	6,8	4,1	1,9	0,6	4,2
Андрій Садовий	**1,8**	1,1	1,8	1,0	1,9	3,0	2,1
Олександр Шевченко	**1,8**	0,7	0,7	0,9	2,2	3,5	1,3
Олег Тягнибок	**1,4**	0,5	0,7	1,2	1,7	2,3	1,7
Андрій Білецький	**0,5**	0,3	0,3	0,5	1,0	0,3	0,7
ІНШИЙ КАНДИДАТ	**4,3**	4,2	5,7	5,3	3,9	3,0	5,2
ВВ/ ВІДМОВА ВІД ВІДПОВІДІ/ НЕ ЗНАЮ	**23,4**	20,4	23,5	24,0	24,2	26,0	17,6
НЕ БРАЛИ Б УЧАСТІ У ГОЛОСУВАННІ	**15,0**	19,7	19,4	14,4	12,6	12,7	12,9

Рисунок 3.13- Результати дослідження

Рисунок 3.14- Результати дослідження

Рисунок 3.15- Результати дослідження

Прогноз. На Вашу думку, хто з даних кандидатів переможе? (%)

100% по кожному стовпчику	Україна загалом	Південь	Схід	Центр Сх	Центр Зх	Захід	Київ
Юлія Тимошенко	**15,4**	12,9	15,2	15,1	18,4	14,1	18,8
Петро Порошенко	**13,4**	9,0	13,6	9,6	15,3	15,7	21,8
Володимир Зеленський	**3,8**	4,8	4,0	4,9	2,7	2,6	4,1
Юрій Бойко	**3,6**	7,1	6,7	4,5	1,2	0,9	2,5
Анатолій Гриценко	**3,3**	2,2	1,3	3,7	2,9	4,9	4,4
Олег Ляшко	**2,7**	2,9	2,2	3,4	3,4	2,0	1,8
Святослав Вакарчук	**2,3**	0,9	2,4	1,9	2,4	3,6	2,4
Вадим Рабінович	**1,8**	3,8	3,5	2,0	0,9	0,4	1,1
Олександр Шевченко	**1,0**	0,6	0,7	0,3	1,0	2,3	0,7
Андрій Садовий	**0,8**	0,5	1,0	0,6	0,7	1,4	0,3
Олег Тягнибок	**0,5**	0,2	0,2	0,5	1,0	0,4	0,8
Андрій Білецький	**0,3**	0,2	0,2	0,3	0,6	0,1	0,7
ІНШИЙ КАНДИДАТ	**0,9**	1,0	0,5	1,1	1,1	0,6	0,7
ВВ/ ВІДМОВА ВІД ВІДПОВІДІ/ НЕ ЗНАЮ	**50,1**	53,8	48,4	52,2	48,1	51,0	39,8

Рисунок 3.16- Результати дослідження

3.4 Результати алгоритмів машинного навчання (класифікації) на власній вибірці даних. Власний прогноз

Постановка задачі: Була розглянута ситуація другого туру президентських виборів між кандидатом А та кандидатом В. Нехай кандидат А – Петро Порошенко, а кандидат В – Юлія Тимошенко

Збір даних: Для вирішення задачі класифікації була створена вибірка даних, де були зібрані думки (короткі речення з негативним або позитивним ставлення до одного з кандидатів) ЗМІ та користувачів щодо обох кандидатів. Думки, що мали позитивний посил щодо кандидата А або негативний посил щодо кандидата В були класифіковані як ‘1’. Думки, що мали позитивний посил щодо кандидата В або негативний посил щодо кандидата А були класифіковані як ‘0’. Також, для подальшого аналізу були зафіксовані дати, коли ці думки були опубліковані.

Результати розв’язання задачі: На прикладі вибірки, яка було сформована за допомогою дослідження ЗМІ і коментарів користувачів, були проаналізовані результати роботи 6 методів машинного навчання: логістична регресія, SVM, KNN, Наївний Байес, дерева ухвалення рішень та алгоритм Random Forest. Для навчання алгоритмів та побудови моделей вибірка була поділена на дві частини: вибірка для навчання алгоритмів та тестова вибірка, на якій перевірялась точність класифікації. За результатами класифікації були сформовані матриці помилок для кожного алгоритму, за якими були визначені наступні показники точності (Точність = Вірно класифіковані думки / Кількість всіх думок у тестовій виборці):

Логістична регресія – 68%

KNN – 64%

SVM - 72%

Наївний Байес – 60 %

Дерево рішень – 56 %

Random Forest -72%

.

Показник детальності і повноти (Детальність = це частка знайдених думок, які мають відповідні до запиту; Повнота = це частка релевантних думок, які успішно знайдені системою пошуку відносно загальної кількості релевантних думок)

7) Логістична регресія: Детальність – 83%, Повнота – 75%
8) KNN: Детальність – 78%, Повнота – 74%
9) SVM : Детальність – 78%, Повнота – 82%
10) Наївний Байес : Детальність – 61%, Повнота – 79%
11) Дерево рішень : Детальність – 61%, Повнота – 73%
12) Random Forest : Детальність – 89%, Повнота – 76%

Для всіх алгоритмів обиралися параметри за замовчуванням аби перевірити їх точність у порівнянні з іншими алгоритмами в однакових умовах. У підсумку, найкращими алгоритмами для вирішення нашої задачі виявилися алгоритми Random Forest та логістична регресія.

Також, за допомогою отриманої вибірки були проаналізовані позитивні та негативні згадування щодо кандидатів А та В за 3 місяці (вересень, жовтень та листопад 2018 року). Результати на Рисунку 3.17.

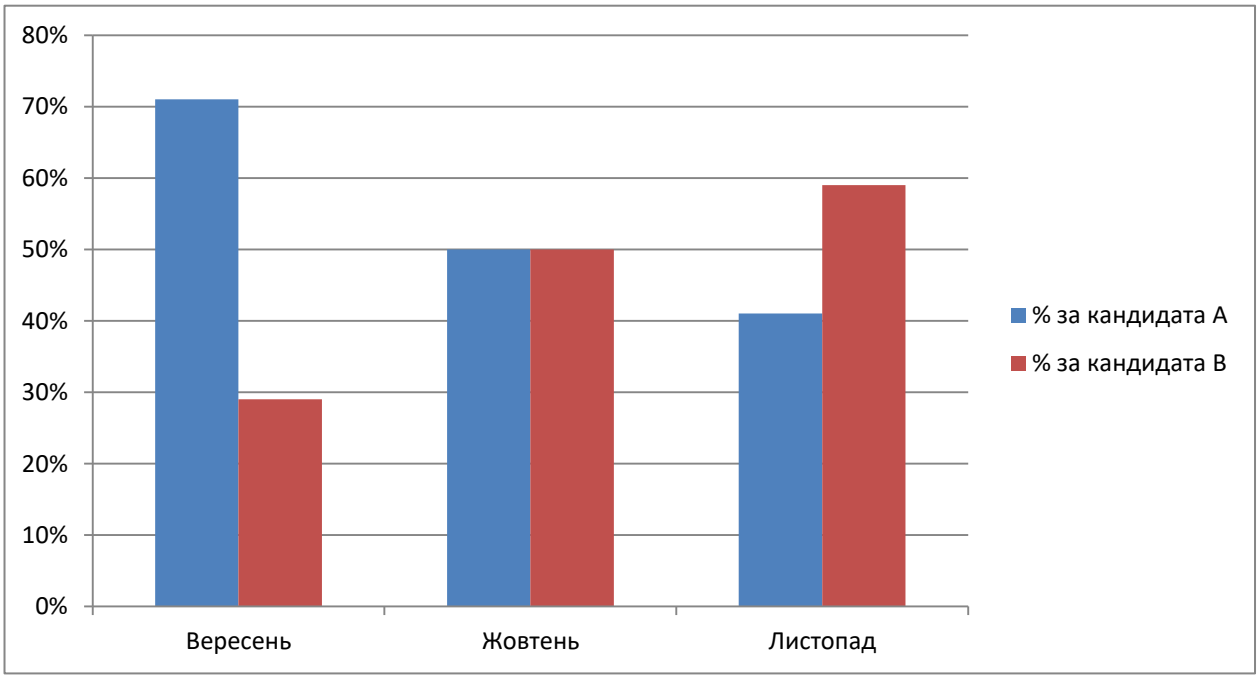

Рисунок 3.17. Рейтинг кандидатів

.

З отриманих даних можна стверджувати, що на початку осені ситуація у протистоянні двох кандидатів була більш менш рівною, якщо ще зважати на невелику кількість згадувань кандидата А та В в ЗМІ та соціальних мережах. Починаючи з листопада, коли обидва кандидати розгорнули передвиборчу кампанію, більшість ЗМІ почали говорити про кандидата В як імовірного переможця у парі, посилаючись на соціальні опитування. Більшість же користувачів соціальних мереж висловлювались більше не на підтримку кандидата А, а проти кандидата В. Якщо ж орієнтуватися на кількість позитивних та негативних згадувань щодо кандидата А та В, то слід вважати, що тенденція зростання рейтингу кандидата В має зберегтися і саме його слід вважати можливим переможцем у другому турі президентських виборів.

Висновки до розділу 3

В даному розділі магістерської дисертації були розглянуті результати соціологічних досліджень щодо електоральний поглядів українців, а також був зроблений власний прогноз переможця другого туру президентських виборів. На виборці, яка було сформована з думок ЗМІ та користувачів, були протестовані алгоритми машинного навчання для класифікації. Найкращим алгоритмом виявися алгоритм Random Forest, який показав найкращу точність класифікації. Також, був зроблений власний прогноз щодо можливого переможця другого туру президентських виборів на основі позитивних та негативних думок користувачів та ЗМІ. Кандидат В (Юлія Тимошенко) має кращі шанси на перемогу ніж кандидат А (Петро Порошенко). Дані результати співпадають з результатами соціологічних опитувань.

.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

Метою цього розділу є опис ідеї стартап-проекту, заснованого на розробках дипломної роботи, аналіз ринкових можливостей, переваги та недоліки даного стартап-проекту, конкурентний аналіз і можливість впровадження.

4.1 Опис ідеї проекту

Прогнозування переможців парламентських або президентських виборів є досить актуальною проблемою для України, особливо в умовах російської агресії. В США методи машинного навчання та аналізу мереж вже більше 10 років застосовуються для прогнозування результатів виборів. Провідні аналітичні компанії весь час отримують замовлення від різних політичних сил для пронозу переможця, оскільки соцопитування не завжди оперативно та об’єктивно показують реальну ситуацію передвиборчих перегонів. В Україні більшість політичних партій та кандидатів досі орієнтуються на результати соцопитувань, для проведення, обробки та аналізу яких потрібна велика кількість часу та фінансових витрат.

Таблиця 4.1- Опис ідеї стартап-проекту

Зміст ідеї	*Напрямки застосування*	*Вимоги для користувача*
Ідея проекту полягає у створенні аналітичної компанії, яка буде надавати консультаційні послуги в області аналізу настроїв користувачів (наприклад для прогнозу переможця президентських виборів)	Прогнозування переможців парламентських та президентських виборів	1) Економічна вигода у порівнянні із проведенням соцопитування 2) Швидший аналіз отриманих результатів, ніж при соцопитуваннях 3) Часова ефективність
	Аналіз настроїв користувачів щодо нового продукту або послуги	
	Аналіз ефективності PR-кампаній	
	Прогнозування результатів ухвалення референдумів, законів і тд.	

Проаналізувавши потенційні техніко-економічні ідеї порівняно з конкурентами, дійшли до висновку, що наразі конкурентів з такими

.

характеристиками, як в нашій ідеї не існує. Наразі, в Україні, використовуються лише соцопитування для прогнозування переможців виборів. Тому провести аналіз з проектами-конкурентами неможливо.

4.2 Технологічний аудит ідеї проекту

Таблиця 4.2- Технологічна здійсненність ідеї проекту

№ п/п	*Ідея проекту*	*Технології її реалізації*	*Наявність технологій*	*Доступність технологій*
1	Створення аналітичної компанії	Власні інвестиції	Наявна	Платна, недоступна
2	Створення аналітичної компанії	Фінансові інвестори	Наявна	Платна, доступна
Обрана технологія реалізації ідеї проекту: Для реалізації ідеї проекту були обрані фінансові інвестори, які зможуть надати необхідні фінансові вливання для створення аналітичної компанії.				

4.3 Аналіз ринкових можливостей для запуску стартап-проекту

Таблиця 4.3- Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, що формує ринок*	*Цільова аудиторія (цільові сегменти ринку)*	*Відмінності у поведінці різних потенційних цільових груп клієнтів*	*Вимоги споживачів до товару*
1	Необхідність прогнозування переможців президентських або парламентських виборів	Політичні партії, кандидати у президенти, їх оточення	Різний підхід до роботи з виборцями	Оперативний аналіз, низькі фінансові витрати у порівнянні із соцопитуваннями
2	Необхідність аналізу настроїв покупців, щодо нового продукту або послуги	Великі компанії	Різні товари та послуги	Оперативний аналіз, низькі фінансові витрати

.

Таблиця 4.4- Фактори загроз

№ п/п	*Фактор*	*Зміст загрози*	*Можлива реакція компанії*
1	Конкуренція	Вихід на ринок великої компанії	Вихід з ринку, запропонувати великій компанії поглинути себе
2	Зміна потреб користувача	Відсутність потреби у послузі	Передбачити можливість до адаптації і надання нових послуг

Таблиця 4.5- Фактори можливостей

№ п/п	*Фактор*	*Зміст можливості*	*Можлива реакція компанії*
1	Зростання можливостей потенційних покупців	Ріст зацікавленості до послуги серед інших груп користувачів	Розширення послуг, які надає компанія
2	Зниження довіри до можливих конкурентів	Втрата позицій можливими конкурентами на ринку	Розширення послуг, які надає компанія, монополізація ринку

Таблиця 4.6- Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Вказати тип конкуренції - недосконала	Немає конкурентів на ринку	-
2. За рівнем конкурентної боротьби - міжнародний	Компанії з інших країн	Найняти необхідних спеціалістів аби у майбутньому вийти на міжнародний ринок
3. За галузевою ознакою - внутрішньогалузева	Конкуренти мають розроблені підходи лише для даної галузі	Адаптувати аналітичний підхід для інших сегментів ринку
4. Конкуренція за видами товарів: - товарно-видова	Види послуг та товарів є однаковими	Створити аналітичний підхід, що демонструє кращі результати, ніж у конкурентів
5. За характером конкурентних переваг - цінова і нецінова	Адаптація технологій аналізу для інших сегментів ринку	Використання менш дорогих технологій у порівнянні із конкурентами
6. За інтенсивністю - марочна	Інші бренди присутні на ринку	Активна реклама, яка вказує на переваги саме даного рішення перед недоліками конкурентів

.

Таблиця 4.7- SWOT-аналіз стартап-проекту

Сильні сторони: Аналітичний підхід, який економніший з точки зору часу та фінансових інвестицій	Слабкі сторони: Ризиковані фінансові інвестиції з боку інвесторів
Можливості: Підвищувати ефективність прогнозу	Загрози: Поява конкурентів з удосконаленими методами аналізу

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.8- Вибір цільових груп потенційних споживачів

№ п/п	*Опис профілю цільової групи потенційних клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
1	Політичні партії, лідери передвиборчої кампанії	Критичним є отримання результату в короткі часові проміжки, робота з результатами в реальному часі	Високий	-	Маючі цінову та часову перевагу у послугах, вийти на ринок не складно
2	Великі компанії	Критичним є отримання результату в короткі часові проміжки, робота з результатами в реальному часі	Середній	-	Маючі цінову та часову перевагу у послугах, вийти на ринок не складно

Таблиця 4.9- Визначення базової стратегії розвитку

№ п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку**
1	Надання консультаційних послуг	Ринкове позиціонування	Точність прогнозу, результат в коротші проміжки часу, можливість адаптації до різних задач	Диференціації

.

Таблиця 4.10- Визначення базової стратегії конкурентної поведінки

№ п/п	*Чи є проект «першопрохідцем» на ринку?*	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки**
1	Так	Так	Ні	Зайняття конкурентної ніші

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.11- Визначення ключових переваг концепції потенційного товару

№ п/п	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити*
1	Результати за короткий проміжок часу	Результати обробки даних алгоритмами можуть бути надані протягом 24 годин	Немає затримки у часі при зборі результатів соцопитувань, їх обробки і тд. Дані вибираються із Інтернету, ЗМІ та соціальних мереж.
2	Релевантний прогноз	Отриманий прогноз буде мати більшу релевантність, ніж соцопитування	Оскільки дані надходять фактично в режимі реального часу, тому і прогноз буде відображати реальну ситуацію, наприклад у передвиборчих перегонах.

Таблиця 4.12. Опис трьох рівнів моделі товару [19].

Рівні товару/послуги	*Сутність та складові*
I. Товар/послуга за задумом	Аналітичний прогноз переможця президентських виборів на основі інформації зі ЗМІ та соціальних мереж із затримкою близько 24 годин.
II. Товар/послуга у реальному виконанні	Властивості/характеристики 1. Прогноз переможця президентських перегонів на основі доступної інформації 2. Наведена методика розрахунку і точність моделі 3. Апробація результатів
	Якість: поділ вибірки на тренувальну та тестову, апробація результатів за допомогою експертних оцінок
	Марка: “Моя компанія”, консультаційні послуги
III. Товар/послуга із підкріпленням	Відсутня підтримка до продажу
	Підтримка для користувачів після продажу згідно з договором
За рахунок чого потенційний товар буде захищено від копіювання: ноу-хау	

.

Висновки до розділу 4

В даному розділі магістерської дисертації був розглянутий підхід щодо реалізації стартап-проекту на основі дослідження та методик, розглянутих у першому, другому та третьому розділі магістерської дисертації. Була наведена ідея стартап-проекту, ринкові можливості, розроблена маркетингова та ринкова стратегії стартап-проекту.

.

ВИСНОВКИ

В процесі виконання даної магістерської дисертації було детально вивчено та ретельно розібрано існуючі методи машинного навчання для класифікації даних, результати їх застосування на вибірках даних. Була розроблена програмна реалізація 6 класифікаторів машинного навчання.

Розглянута математична модель “мішок слів”, яка застосовується для аналізу тональності тексту. Розібрані приклади із застосування даної моделі для вирішення задачі класифікації позитивних та негативних відгуків на ресторан, позитивних та негативних думок щодо кандидатів у президенти.

Представлено результати роботи моделі “мішок слів” та методів машинного навчання для вирішення задачі прогнозування переможця президентських перегонів. Найкращі результати продемонстрували алгоритми Random Forest (Точність -72%, Детальність – 89%, Повнота – 76%), метод опорних векторів (SVM) (Точність -72%, Детальність – 78%, Повнота – 82%) та логістична регресія (Точність -68%, Детальність – 83%, Повнота – 75%).

Запропонований підхід дозволяє прогнозувати переможця президентських виборів у другому турі з меншими фінансовими і часовими витратами, оскільки не має потреби у проведенні соціологічних опитувань. Результати пронозу за допомогою аналізу тональності думок можна отримати набагато швидше, оскільки не потрібно багато часу на збір та аналіз отриманої інформації. Маючи велику вибірку даних точність класифікації алгоритмів машинного навчання можна довести більш ніж до 90%.

.

ПЕРЕЛІК ДЖЕРЕЛ, ПОСИЛАНЬ

1 Математические модели текста [Електронний ресурс] – Режим доступу до ресурсу: http://lab314.brsu.by/kmp-lite/kmp2/JOB/CModel/BoW-Q.htm

2 Классификация текстов с помощью мешка слов. Руководство [Електронний ресурс] – Режим доступу до ресурсу: http://datareview.info/article/klassifikatsiya-tekstov-s-pomoshhyu-meshka-slov-rukovodstvo/ - 08.07.2015 р.

3 МОНІТОРИНГ ЕЛЕКТОРАЛЬНИХ НАСТРОЇВ УКРАЇНЦІВ [Електронний ресурс] – Режим доступу до ресурсу: https://www.kiis.com.ua/?lang=ukr&cat=reports&id=800&page=1_-13.11.2018 р.

4 Соціально політична ситуація в Україні [Електронний ресурс] – Режим доступу до ресурсу: http://razumkov.org.ua/uploads/socio/2018_Press_release_september.pdf

5 MACHINE LEARNING A-Z™: DOWNLOAD PRACTICE DATASETS [Електронний ресурс] – Режим доступу до ресурсу: https://www.superdatascience.com/machine-learning/

6 Логістична регресія. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Логістична_регресія

7 Метод k-найближчих сусідів. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Метод_k-найближчих_сусідів

8 Метод k-ближайших соседей. Википедия. [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Метод_k-ближайших_соседей

9 Метод опорних векторів. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Метод_опорних_векторів

.

10 Наївний баєсів класифікатор. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Наївний_баєсів_класифікатор

11 Наивный байесовский классификатор. Википедия [Електронний ресурс] – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Наивный_байесовский_классификатор

12 Дерево ухвалення рішень. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Дерево_ухвалення_рішень

13 Random Forest. Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Random_forest

14 Practical Statistics for Data Scientists / П. Брюс, Є. Брюс., 2018. – 304 с.

15 Machine Learning Algorithms: A reference guide to popular algorithms for data science and machine learning / Giuseppe Bonaccorso., 2017. – 360 с. – (Packt Publishing).

16 Визначення місця і ролі національної єдності у Стратегії національної безпеки України, програмах політичних партій і передвиборчому процесі 2007 року / А.Б. Качинський, С.П. Герасимчук, Д.І. Остапчук, Л.М. Шипілова., 2008. – 63с. – (Інтертехнологія)

17 Lane H. Natural Language Processing in Action: Understanding, analyzing, and generating text with Python / H. Lane, H. Hapke, C. Howard., 2018. – 420 с. – (Manning Publications).

18 Попередній огляд свіжого соцопитування MPI: передвиборчі настрої в Україні [Електронний ресурс] – Режим доступу до ресурсу: https://www.iri.org.ua/novini/poperedniy-oglyad-svizhogo-socopituvannya-mri-peredviborchi-nastroi-v-ukraini - 09.11.2018р.

19 Розроблення стартап-проекту: Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей/ О. А. Гавриш., 2016 - НТУУ «КПІ».

.

ДОДАТОК А

Нижче наведено програмну реалізацію методів, які були описані у першій та другій частині магістерської роботи. Всі різні класифікатори наведено в одній частині програми, але для коректної роботи їх потрібно виконувати по черзі. Для навчання алгоритмів була обрана вибірка, що складається з 1000 розмічених відгуків користувачів про ресторан – негативних та позитивних, а також власна вибірка позитивних та негативних думок стосовно кандидатів у президенти А та В.

```
 # Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)

# Cleaning the texts
import re
import nltk
#nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
corpus = []
for i in range(0, 1000):
    review = re.sub('[^a-zA-Z]', ' ', dataset['Review'][i])
    review = review.lower()
    review = review.split()
    ps = PorterStemmer()
```

.

```
    review = [ps.stem(word) for word in review if not word in
set(stopwords.words('english'))]
    review = ' '.join(review)
    corpus.append(review)

# Creating the Bag of Words model
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features = 1500)
X = cv.fit_transform(corpus).toarray()
y = dataset.iloc[:, 1].values

# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state =
0)

# Fitting different classifiers to the Training set

#Fitting Decision Tree classifier to the Training set
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)

# Fitting KNN classifier to the Training set
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)

# Fitting Logistic regression classifier to the Training set
from sklearn.linear_model import LogisticRegression
```

.

```
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)

# Fitting Naive Bayes to the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, y_train)

# Fitting Random Forest classifier to the Training set
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier.fit(X_train, y_train)

# Fitting SVM classifier to the Training set
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, y_train)



# Predicting the Test set results
y_pred = classifier.predict(X_test)

# Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
```

.